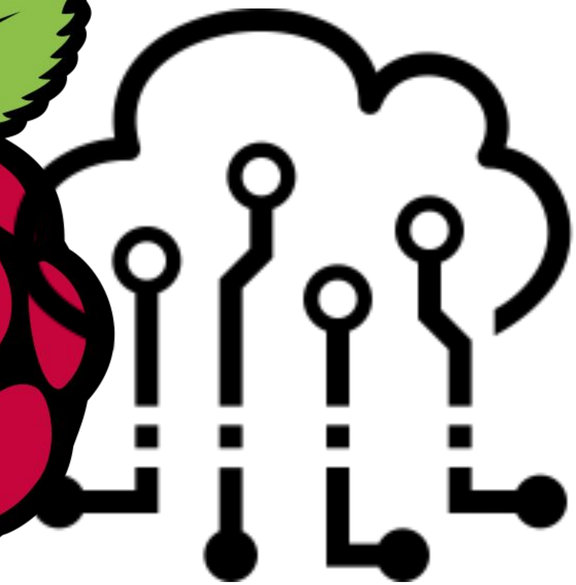
Internet of Things (IoT) Management with Raspberry Pi (Raspberry Pico W) - v6



Trainer: Man Guo Chang

Website:www.tertiarycourses.com.sg
Email: enquiry@tertiaryinfotech.com

About the Trainer

Mr. Man Guo Chang graduated from Nanyang Technological University, School of Electrical and Electronic Engineering, major in Computer Engineering.

Mr. Man has more than 25 years of working experience in the Semiconductor field, specialized in IC Testing, Product Engineering, Data Analysis, and Software Development.

Mr. Man is an ACTA certified trainer. His skill set includes Website Development, Software Development, Machine Vision, Internet of Things, ROS, Cyber Security, etc.

Let's Know Each Other...

Say a bit about yourself
- Name
- What Industry you are from?
- Do you have any prior knowledge in IoT
- Why do you want to learn IoT?
- What do you want to learn from this course?

Ground Rules

- Set your mobile phone to silent mode
- Actively participate in the class. No question is stupid.
- Respect each other view
- Exit the class silently if you need to step out for phone call, toilet break

Ground Rules for Virtual Training

- Upon entering, mute your mic and turn on the video. Use a headset if you can
- Use the 'raise hand' function to indicate when you want to speak
- Participant actively. Feel free to ask questions on the chat whenever.
- Facilitators can use breakout rooms for private sessions.

WSQ and SSG TG Application Form

Please fill up the following WSQ and SSG TG Application Form for TRACOM survey, e-cert generation and WSQ funding

https://forms.gle/pJ2WxHZ3fyRbDLVu6

Digital Attendance

- It is <u>mandatory</u> for you to take both AM and PM digital attendance especially for WSQ courses.

- The trainer or administrator will show you the digital attendance QR code generated from TPG.

- Please use your own QR code scanner to scan the QR code, login to your SingPass and submit your attendance.

Guidelines for Facilitators

1. Once all the participants are in and introduce themselves
2. Go to gallery mode, take a snapshot of the class photo - makes sure capture the date and time
3. Start the video recording (only for WSQ courses)
4. Continue the class
5. Before the class end on that day, take another snapshot of the class photo - makes sure capture the date and time
6. For NRIC verification, facilitator to create breakout room for individual participant to check (only for WSQ courses)
7. Before the assessment start, take another snapshot of the class photo - makes sure capture the date and time (only for WSQ courses)
8. For Oral Questioning assessment, facilitator to create breakout room for individual participant to OQ (only for WSQ courses)
9. End the video recording and upload to cloud (only for WSQ courses)
10. Assessor to send all the assessment records, assessment plan and photo and video to the staff (only for WSQ courses).

Prerequisite

The following prior knowledge is assumed

- Basic Raspberry Pi operation

Learning Outcomes

By end of the course, learners should be able to

- analyze the uses and functions of IoT technologies
- collect environment data and post the data to cloud computing
- use cloud computing to analyse and visualize IoT data
- monitor data and trigger control on devices

Agenda

Topic 1 Overview of Internet of Things (IoT)

- What is IoT?
- Sensors and Actuators for IoT
- Wireless Communication Technologies for IoT
- IoT Applications and Use Cases

Topic 2 Collect and Post Environmental Data to Cloud

- What is Cloud Computing
- Setup Cloud Computing Account
- Collect Environmental Data with Sensors
- Transmit Environmental Data using Raspberry Pi Pico W
- Post Data to Cloud using MQTT API

Agenda

Topic 3 IoT Data Analytics and Visualization

- Analyze IoT Data on Cloud
- Visualize IoT Data on Cloud

Topic 4 Monitor Data to Trigger Control from Cloud

- Read Data using MQTT API
- Trigger Control on Devices
- IoT Security

Final Assessment

- Written Assessment (Q&A)
- Practicum Performance

Download Course Material

- Your can download the course material from Google Classroom
- Enter the class code below to join the class on the top right.
- Go to Classwork >> Course Material
- If you cannot access the google classroom, please inform the trainer or staff.

https://classroom.google.com/c/MTU0MTU1NjIxNjA5?cjc=7xacsqx

7xacsqx

CERTIFICATE

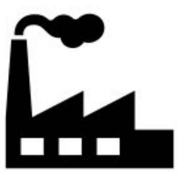
Two e-certificates will be awarded to trainees who have demonstrated competency in the WSQ assessment and achieved at least 75% attendance.

- A SkillsFuture WSQ Statement of Attainment (SOA) issued by WSG. Typically take 4 weeks
- Certification of Completion issued by Tertiary Infotech Pte Ltd, immediately after the course

Topic 1
Overview of Internet of Things (IoT)

4th Industrial Revolution

Manufacturing continues to evolve and has progressed through four stages:
1.0: Steam power to mechanization
2.0: Mass production and powered by electricity
3.0: Computers and automation
4.0: Cyber-physical systems to enable the computerization.



Industry 1.0
Mechanization, steam power, weaving loom

(18th century, 1760 - 1840)

Industry 2.0
Mass production, assembly line, electrical energy

(19th century, 1871 - 1914)

Industry 3.0
Automation, computers and electronics

(late 20th century, 1970 - present)

Industry 4.0
Cyber Physical Systems, internet of things, networks

(early 21st century, 2011 - present)

4th Industrial Revolution



- Emphasizes heavily on automation, machine learning, real-time data, and interconnectivity
- Smart manufacturing
- Synchronizes physical production and operations with Smart Digital Technology

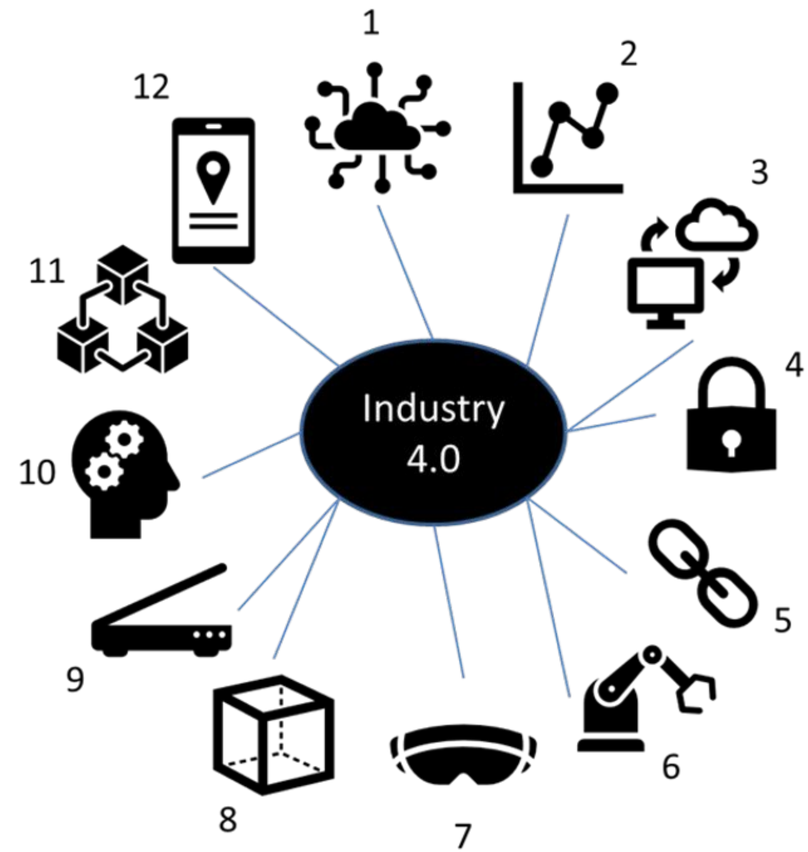
4th Industrial Revolution

- Access and stay connected to real-time insights across products, people, partners, and processes.
- Advanced tools and techniques for improving the efficiency of manufacturing operations
- Revolutionizing the way every single operation of organization.

4th Industrial Revolution

There are many technologies and applications being used in Industry 4.0 to achieve goals and objectives.

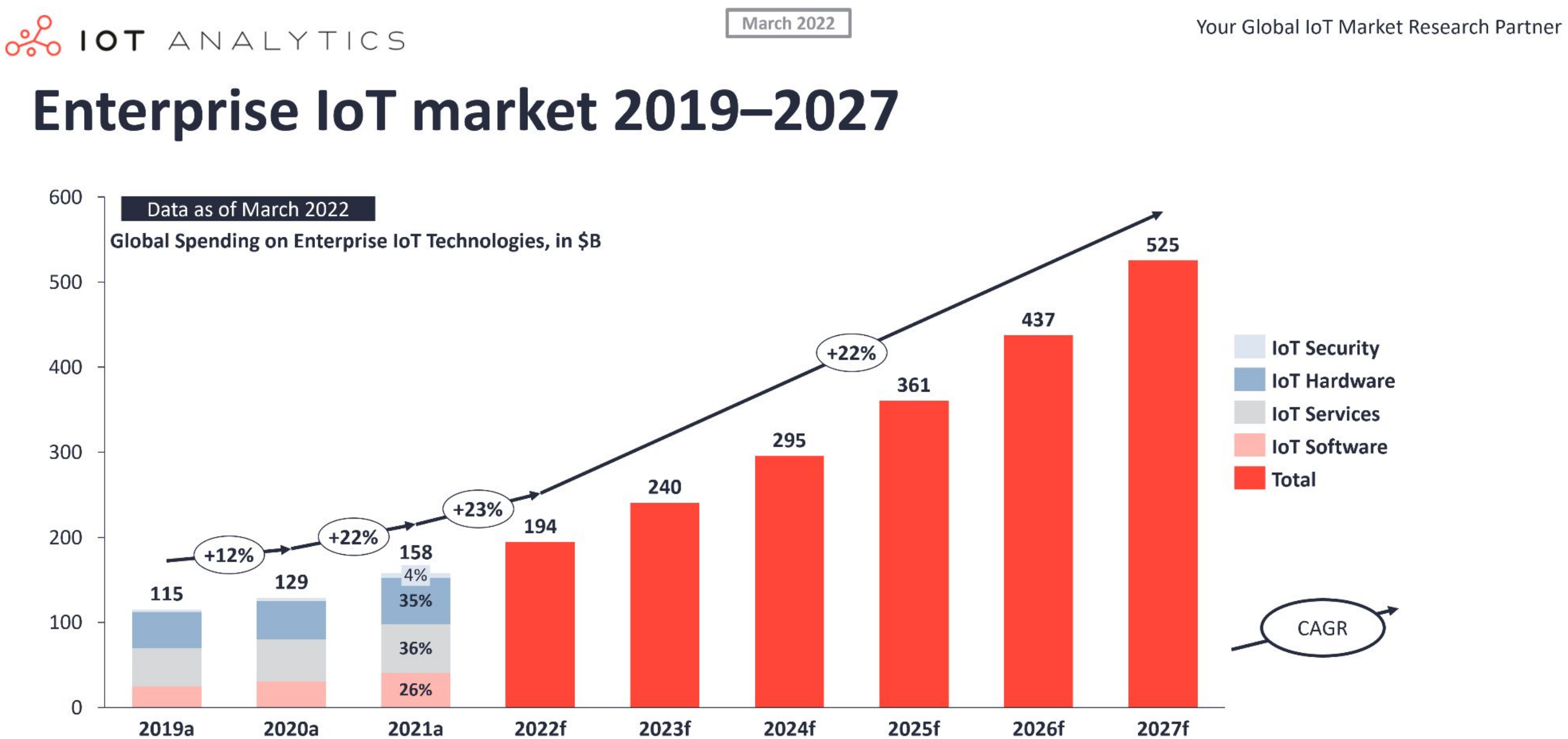
- Industrial Internet of Things (IIoT)
- Big Data and Analytics
- Cloud Computing
- Cybersecurity
- Horizontal & Vertical Integration
- Autonomous Robots
- Augmented Reality
- Additive Manufacturing
- Simulation or Digital Twin
- Artificial Intelligence
- Blockchain Technology
- 5G Technology



Internet of Things (IoT)

IoT (Internet of Things) refers to the network of physical devices, vehicles, buildings, and other objects that are embedded with sensors, software, and connectivity, allowing them to collect and exchange data with each other and with central servers or the cloud.

Internet of Things (IoT)



Note: IoT Analytics defines IoT as a network of internet-enabled physical objects. Objects that become internet-enabled (IoT devices) typically interact via embedded systems, some form of network communication, or a combination of edge and cloud computing. The data from IoT-connected devices is often used to create novel end-user applications. Connected personal computers, tablets, and smartphones are not considered IoT, although these may be part of the solution setup. Devices connected via extremely simple connectivity methods, such as radio frequency identification or quick response codes, are not considered IoT devices. a: Actuals, f: Forecast

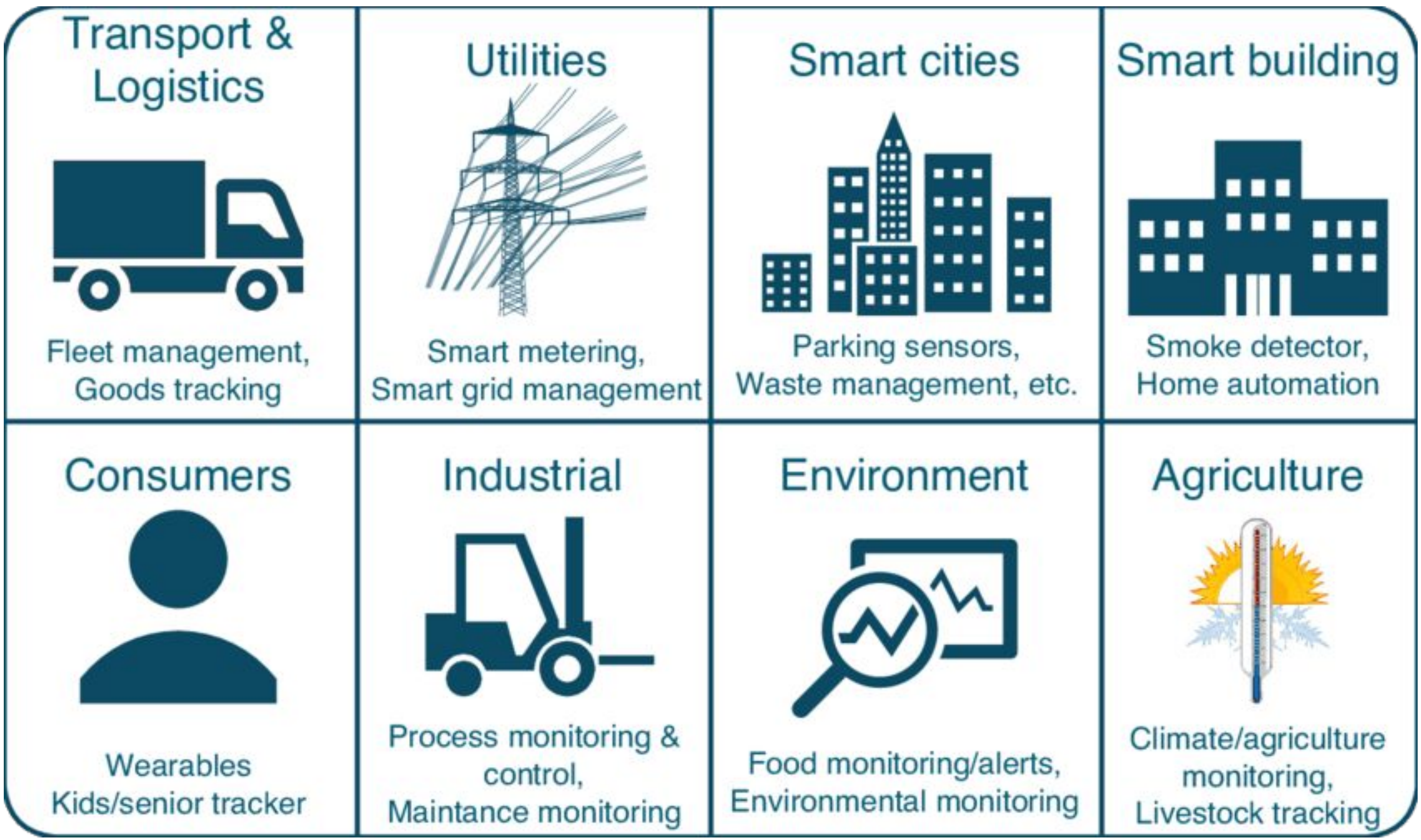
Source: IoT Analytics Research 2022. We welcome republishing of images but ask for source citation with a link to the original post or company website.

Driving Forces of IoT

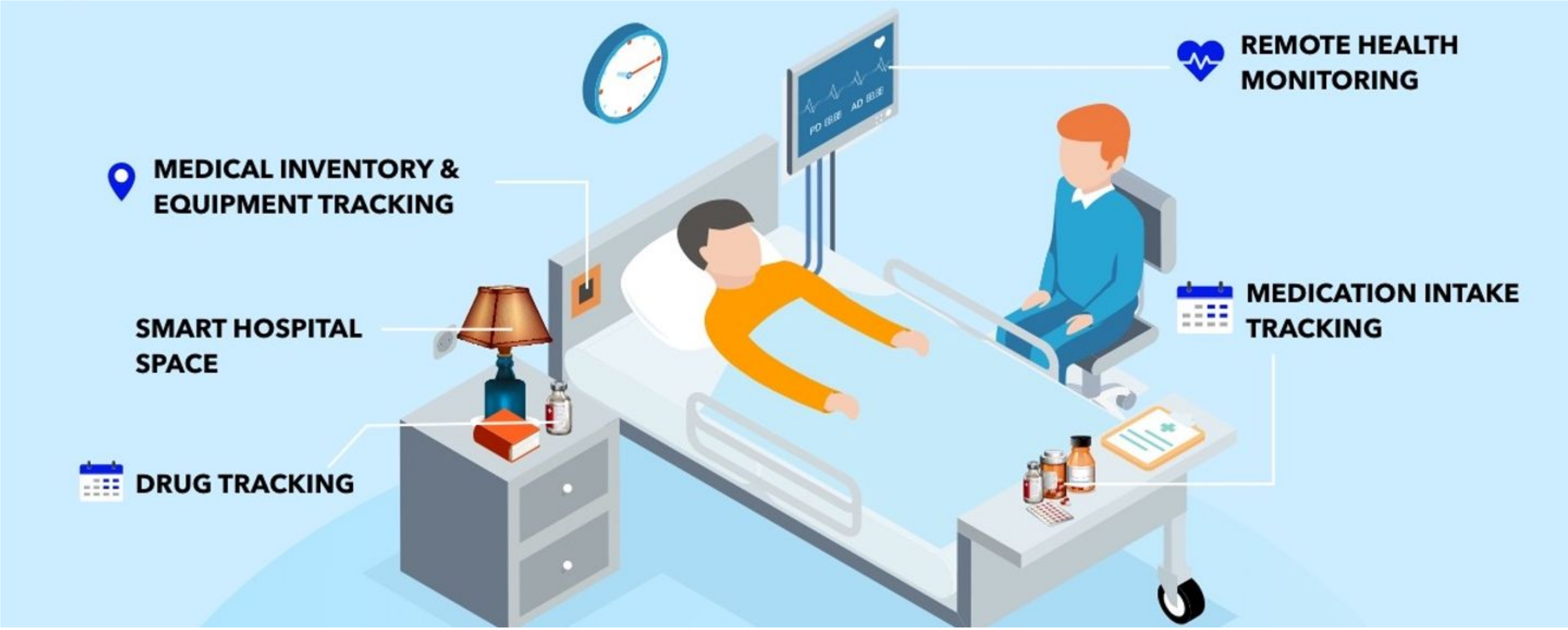
- Low cost sensors and long range wireless technology are instrumental to IoT
- 5G will drive IoT to mass adoption.

IoT Applications

IoT based application has found its way into every conceivable field, from agriculture to aerospace



IoT Use Case - Healthcare



IoT Use Case - Healthcare

- Doctors, nurses, and orderlies often need to know the exact location of patient-assistance assets such as wheelchairs.
- When a hospital's wheelchairs are equipped with IoT sensors, they can be tracked from the IoT asset-monitoring application so that anyone looking for one can quickly find the nearest available wheelchair.
- Many hospital assets can be tracked this way to ensure proper usage as well as financial accounting for the physical assets in each department.
-

IoT Use Case - Agriculture

Productivity

Crop Yield Maximization

Preventative maintenance

Disrupting the mating patterns of pests

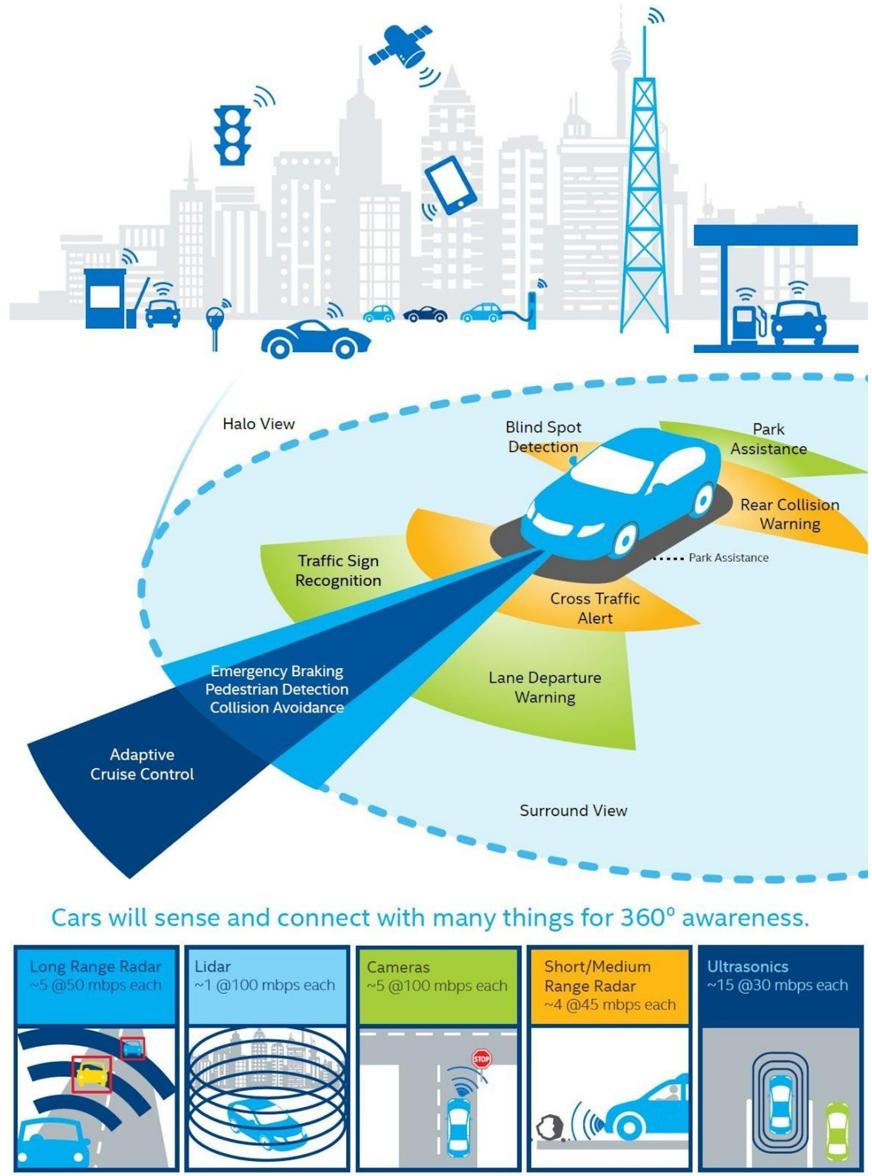
Preservation

Drought response, minimize waste

End-to-End lifecycle monitoring

Tracking product lifecycle in real-time

IoT Use Case - Connected Vehicles



IoT Use Case - Connected Vehicles

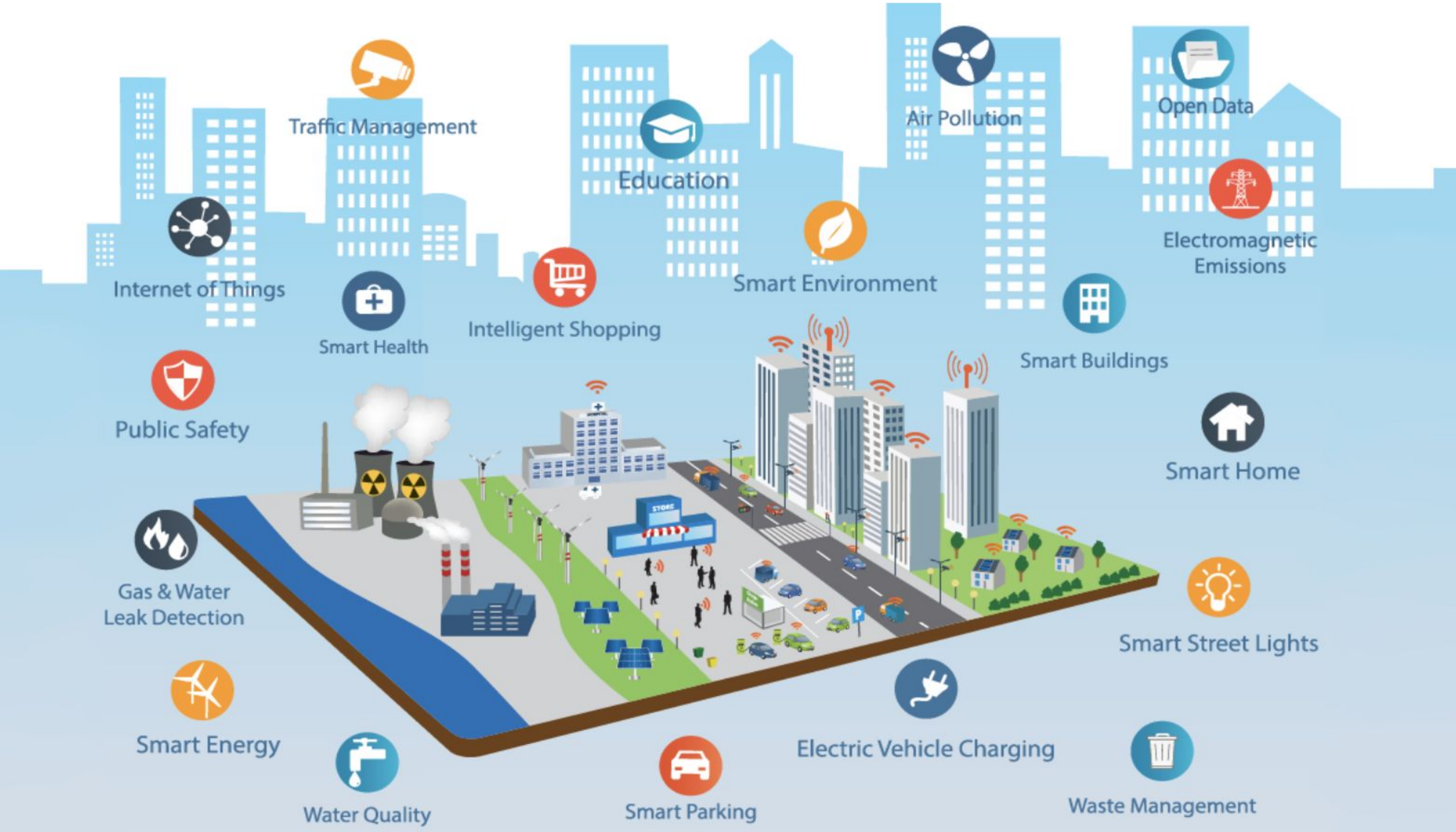
- There are many ways vehicles, such as cars, can be connected to the internet. It can be through smart dashcams, infotainment systems, or even the vehicle's connected gateway. They collect data from the accelerator, brakes, speedometer, odometer, wheels, and fuel tanks to monitor both driver performance and vehicle health.
- Connected cars have a range of uses:
  - Monitoring rental car fleets to increase fuel efficiency and reduce costs.
  - Helping parents track the driving behavior of their children.
  - Notifying friends and family automatically in case of a car crash.
  - Predicting and preventing vehicle maintenance needs.

IoT Use Case - Smart Home

IoT Use Case - Smart Home

- Smart home devices are mainly focused on improving the efficiency and safety of the house, as well as improving home networking. Devices like smart outlets monitor electricity usage and smart thermostats provide better temperature control.
- Home security systems like door locks, security cameras, and water leak detectors can detect and prevent threats, and send alerts to homeowners.
- Connected devices for the home can be used for:
    - Automatically turning off devices not being used.
    - Rental property management and maintenance.
    - Finding misplaced items like keys or wallets.
    - Automating daily tasks like vacuuming, making coffee, etc.

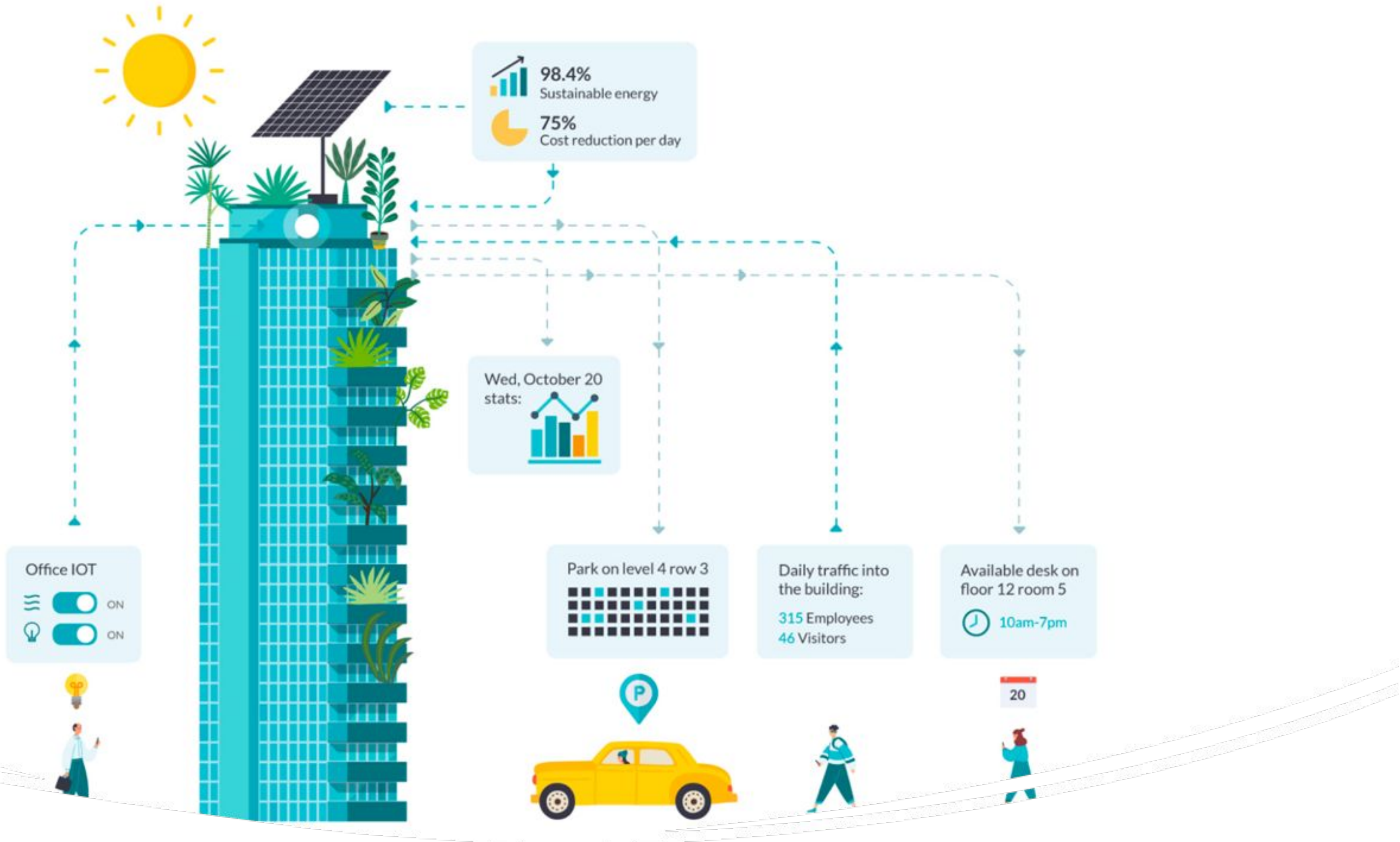
Use Case - Smart City



IoT Use Case - Smart Home

- IoT applications have made urban planning and infrastructure maintenance more efficient. Governments are using IoT applications to tackle problems in infrastructure, health, and the environment.
- IoT applications can be used for:
  - Measuring air quality and radiation levels.
  - Reducing energy bills with smart lighting systems.
  - Detecting maintenance needs for critical infrastructures such as streets, bridges, and pipelines.
  - Increasing profits through efficient parking management.

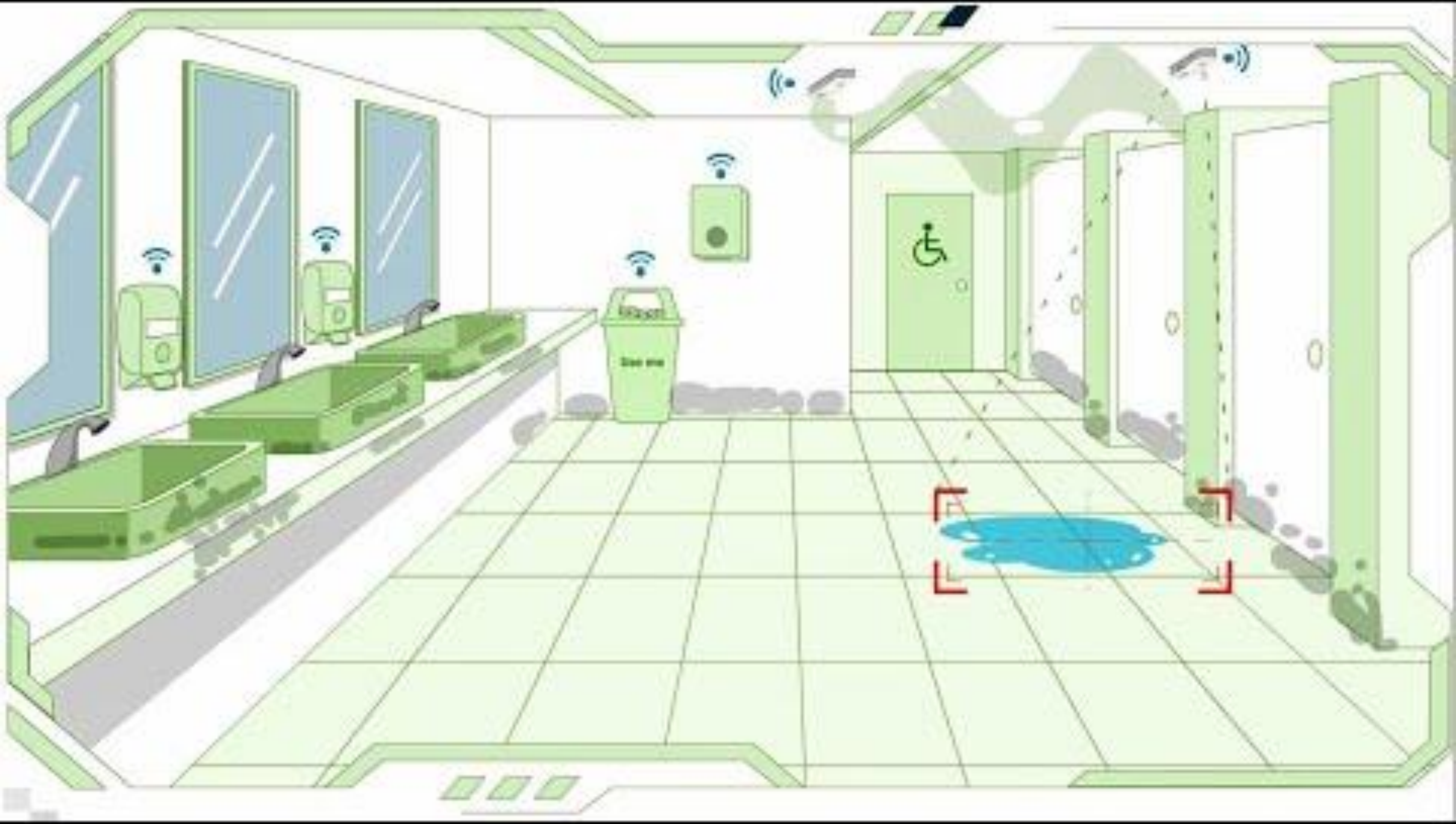
IoT Use Case - Smart Building



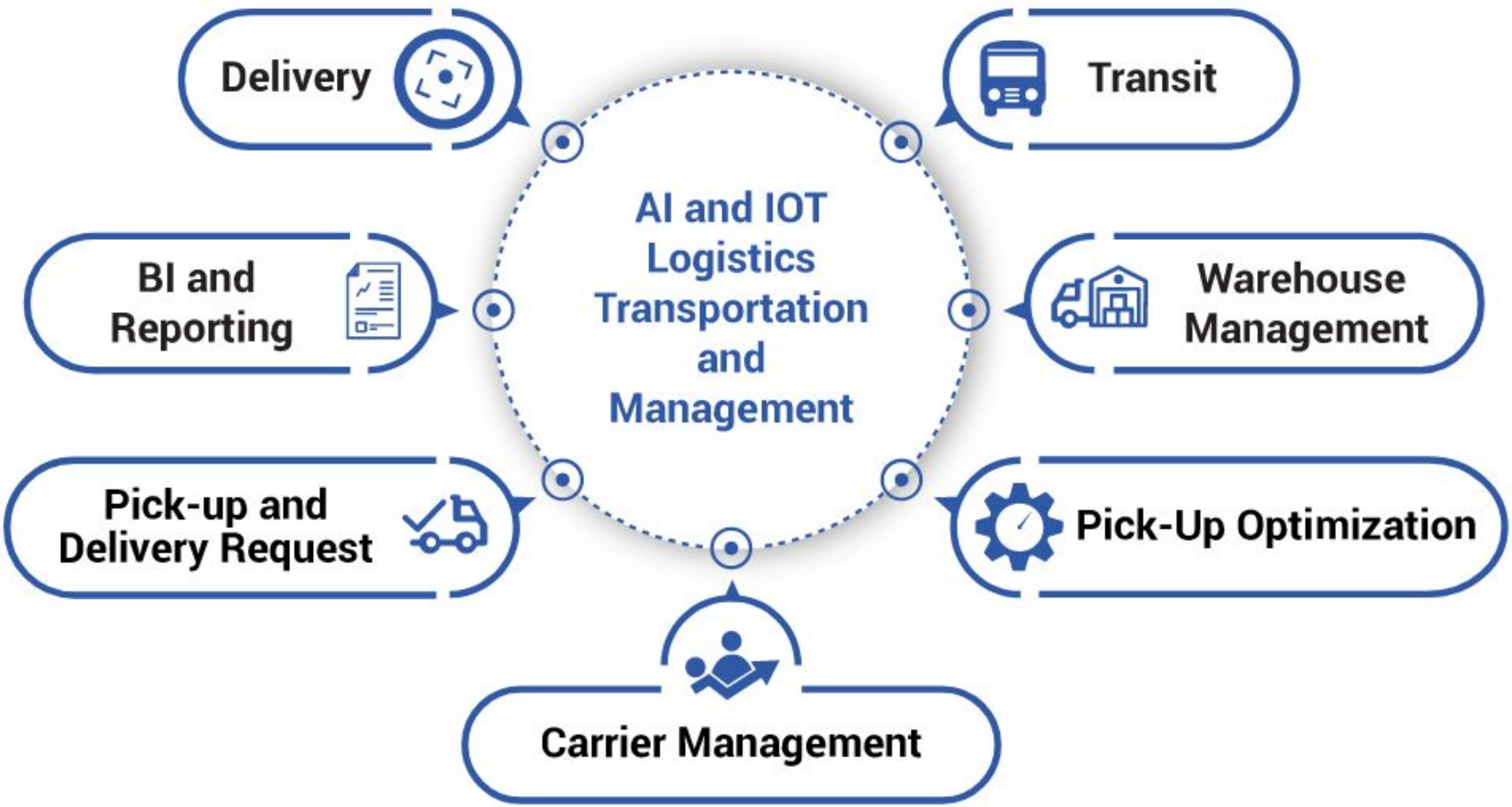
IoT Use Case - Smart Building

- Buildings such as college campuses and commercial buildings use IoT applications to drive greater operational efficiencies.
- IoT devices can be use in smart buildings for:
    - Reducing energy consumption.
    - Lowering maintenance costs.
    - Utilizing work-spaces more efficiently.

IoT Use Case - Smart Toilet



IoT Use Case - Logistics



IoT Use Case - Logistics

- Fleets of cars, trucks, ships, and trains that carry inventory can be rerouted based on weather conditions, vehicle availability, or driver availability, thanks to IoT sensor data.
- The inventory itself could also be equipped with sensors for track-and-trace and temperature-control monitoring.
- The food and beverage, flower, and pharmaceutical industries often carry temperature-sensitive inventory that would benefit greatly from IoT monitoring applications that send alerts when temperatures rise or fall to a level that threatens the product.

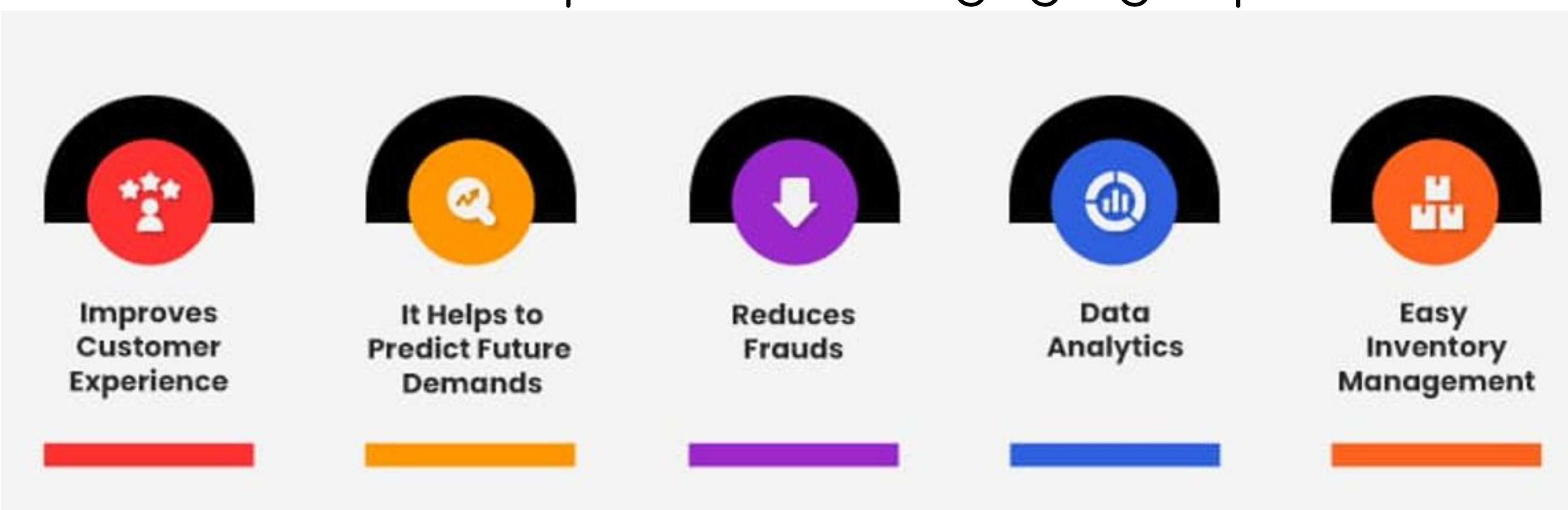
IoT Use Case - Manufacturing

IoT Use Case - Manufacturing

- Manufacturers can gain a competitive advantage by using production-line monitoring to enable proactive maintenance on equipment when sensors detect an impending failure.
- Sensors can measure when production output is compromised. With the help of sensor alerts, manufacturers can quickly check equipment for accuracy or remove it from production until it is repaired.
- This allows companies to reduce operating costs, get better uptime, and improve asset performance management.

IoT Use Case - Retail

- IoT applications allow retail companies to manage inventory, improve customer experience, optimize supply chain, and reduce operational costs.
- Smart shelves fitted with weight sensors can collect RFID-based information and send the data to the IoT platform to automatically monitor inventory and trigger alerts if items are running low.
- Beacons can push targeted offers and promotions to customers to provide an engaging experience.

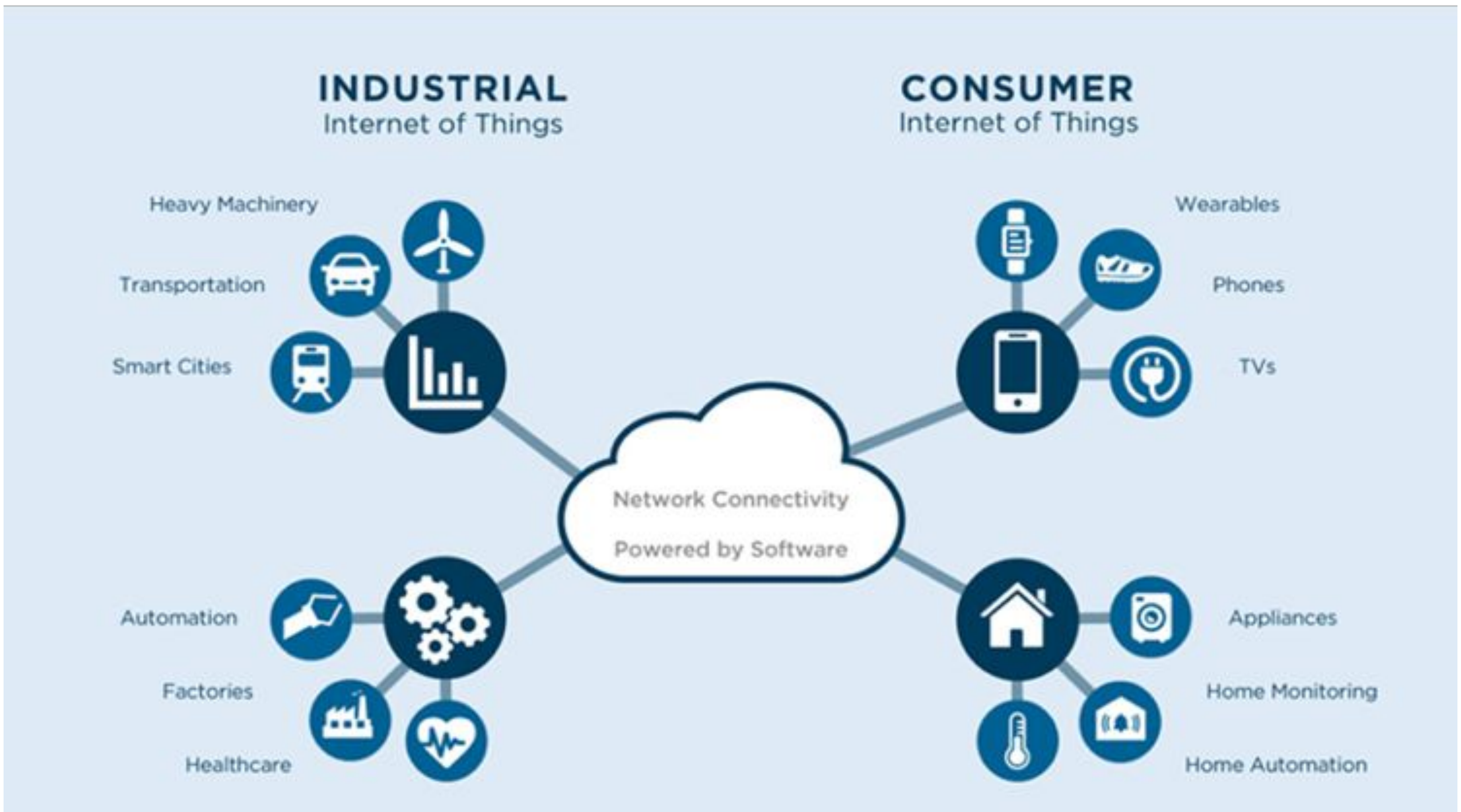


Types of IoT

- **Consumer IoT** - Primarily for everyday use. Eg: home appliances, voice assistance, and light fixtures.
- **Commercial IoT** - Primarily used in the healthcare and transport industries. Eg: smart pacemakers and monitoring systems, connected cars
- **Industrial Internet of Things (IIoT)** - Primarily used with industrial applications, such as in the manufacturing and energy sectors. Eg: Digital control systems, smart agriculture and industrial big data.
- **Infrastructure IoT** - Primarily used for connectivity in smart cities. Eg: infrastructure sensors and management systems.

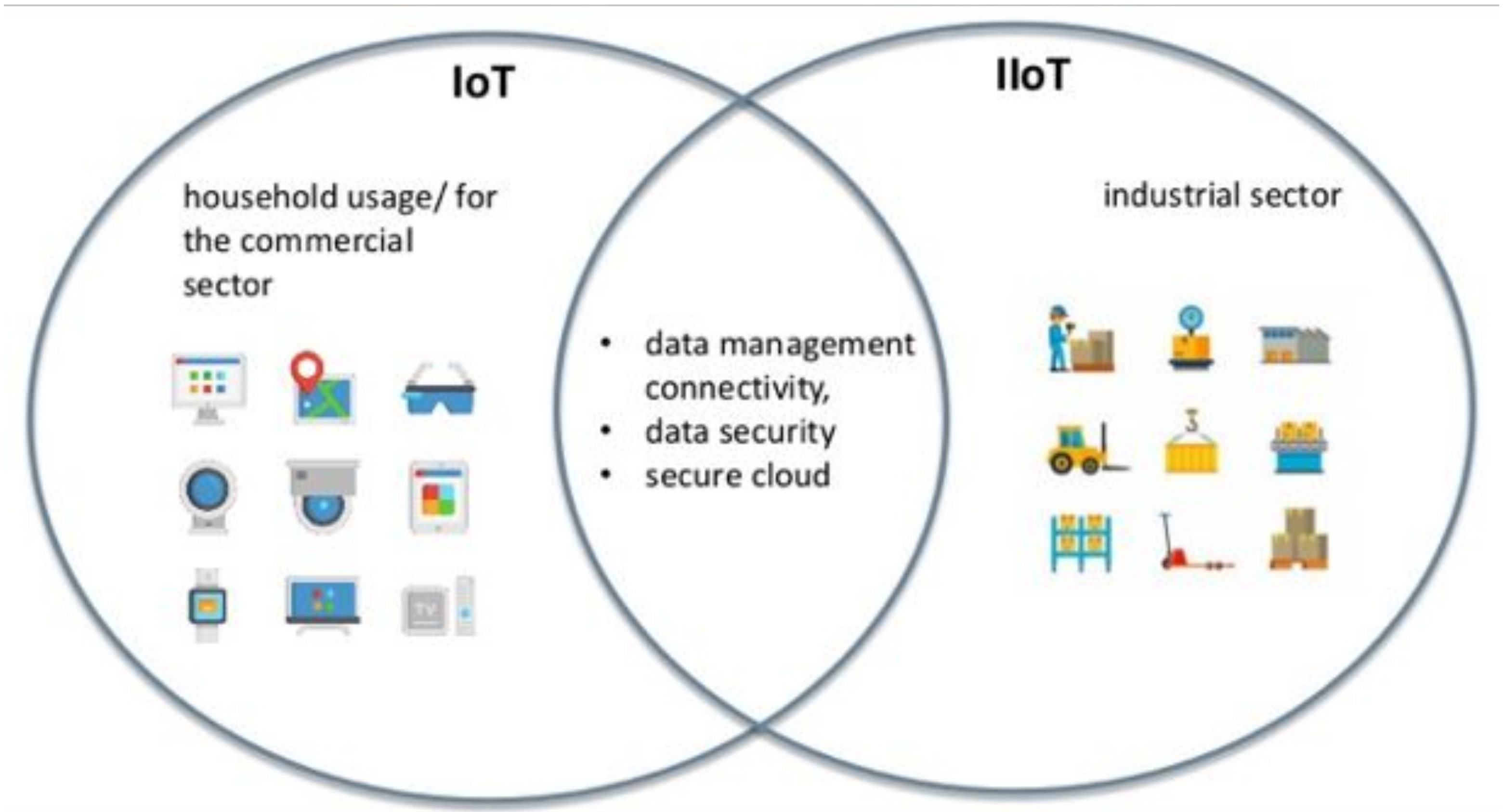
What is Industrial IoT?

The industrial internet of things (IIoT) refers to the extension and use of the internet of things (IoT) in industrial sectors and applications.



IoT vs IIoT

Industrial devices, from sensors to equipment, give business owners detailed, real-time data that can be used to improve business processes.



IoT Implementation

- **Hardware:** IoT devices typically require hardware components such as sensors, processors, and communication modules that can connect to the internet or other networks.
- **Software:** IoT devices also require software that can interpret and process the data collected by the sensors and communicate with other devices or servers.
- **Communication protocols:** IoT devices use different communication protocols, such as Wi-Fi, Bluetooth, Zigbee, and cellular networks, to communicate with each other and with central servers or the cloud.
- **Data analytics:** The data collected by IoT devices needs to be processed and analyzed to derive meaningful insights that can be used for decision-making.

IoT Implementation

- **Security:** IoT devices and networks need to be secured against cyber attacks, data breaches, and other security threats.
- **Cloud computing:** IoT data is often stored and processed in the cloud, which requires knowledge of cloud computing platforms, such as AWS, Azure, or Google Cloud.
- **Artificial Intelligence and Machine Learning:** IoT systems can use AI and ML to improve decision-making, predictive maintenance, and automate tasks.
- **Energy efficiency:** IoT devices often operate on battery power or other low-power sources, which requires knowledge of energy-efficient design and management.

IoT Components

Sensors

Collecting data

Connectivity

Sending data to cloud

Data Processing

Making data useful

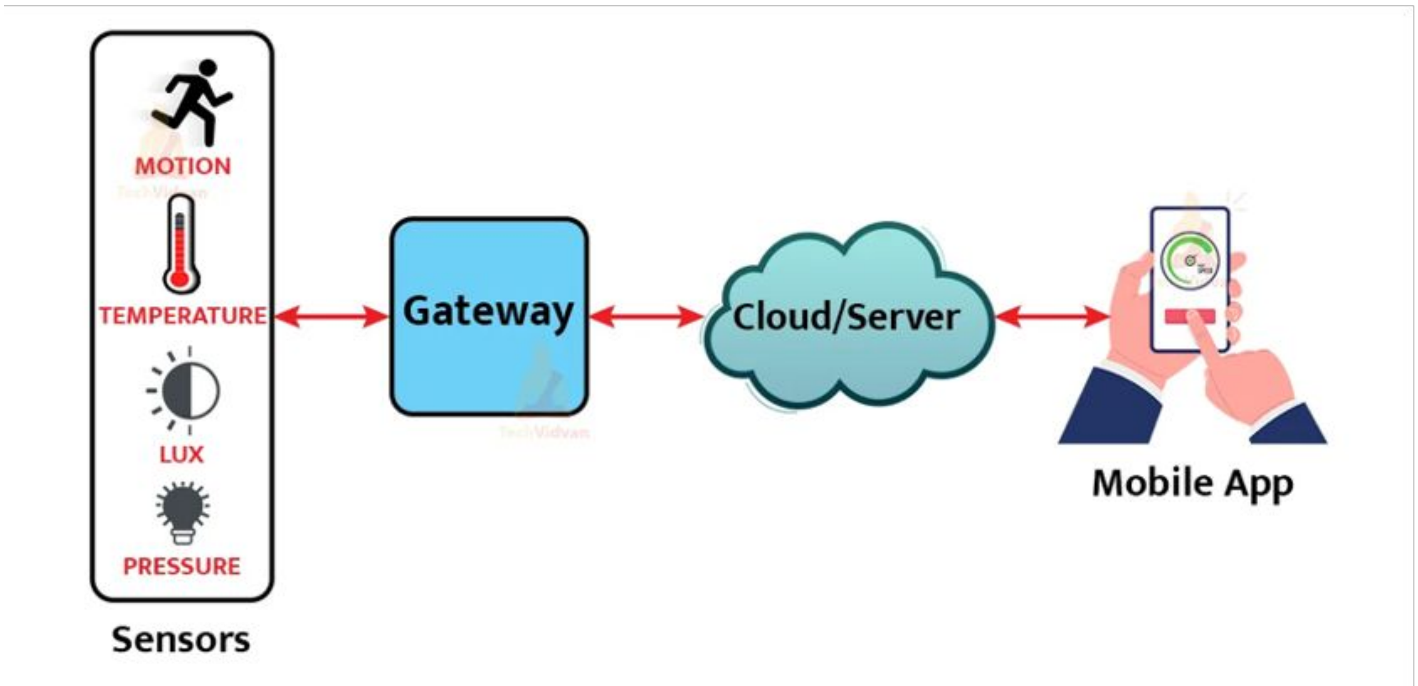
User Interface

Delivering information to user

A “thing” is any object with embedded electronics that can transfer data over a network — without any human interaction.

How does IoT work?

- Devices and objects with built in sensors are connected to an Internet of Things platform, which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs.
- These powerful IoT platforms can pinpoint exactly what information is useful and what can safely be ignored. This information can be used to detect patterns, make recommendations, and detect possible problems before they occur.



What Are IoT Devices?

- IoT devices are hardware devices, such as sensors, gadgets, appliances and other machines that collect and exchange data over the Internet.
- They are programmed for certain applications and can be embedded into other IoT devices.
- For example, an IoT device in your car can identify the traffic ahead and send out a message automatically to the person you are about to meet of your impending delay.

Functionalities of IoT devices

- Different IoT devices have different functions, but they all have similarities in terms of how they work.
- Firstly, IoT devices are physical objects that sense things going on in the physical world. They contain an integrated CPU, network adapter and firmware, and are usually connected to a Dynamic Host Configuration Protocol server. It also requires an IP address to function over the network.
- Most IoT devices are configured and managed through a software application. For example, an app on your smartphone to control the lights in your home.
- Some devices also have integrated web servers, which eliminates the need for external applications. For example, the lights switch on immediately when you enter a room.

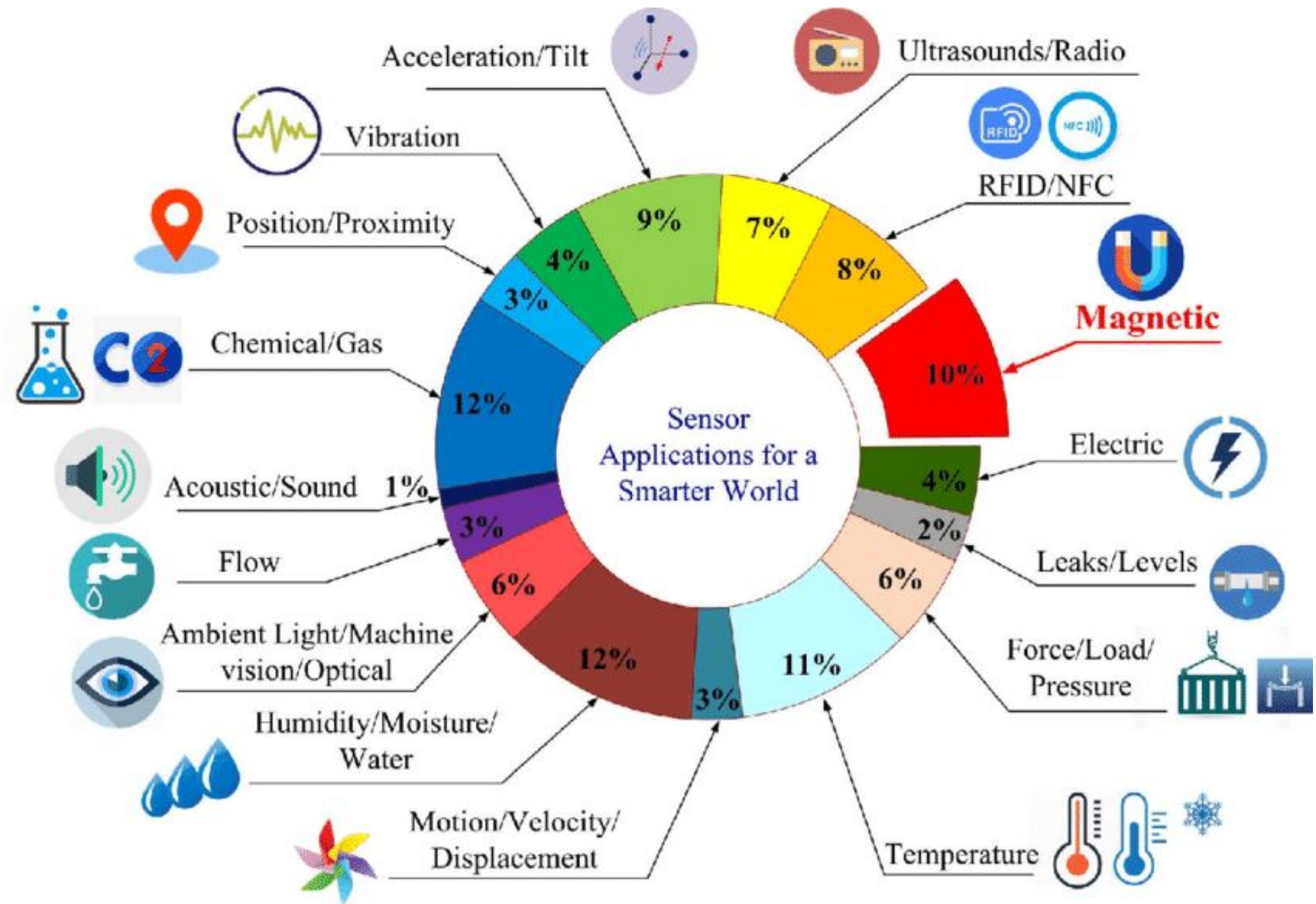
Examples of IoT Devices

- **Home Security** - The key driver behind smart and secure homes is IoT. A variety of sensors, lights, alarms and cameras (all of which can be controlled from a smartphone) are connected via IoT to provide 24x7 security.
- **Activity Trackers** - Smart home security cameras provide alerts and peace of mind. Activity trackers are sensor devices that can monitor and transmit key health indicators in real-time. You can track and manage your blood pressure, appetite, physical movement and oxygen levels.
- **Industrial Security and Safety** - IoT-enabled detection systems, sensors and cameras can be placed in restricted areas to detect trespassers. They can also identify pressure buildups and small leaks of hazardous chemicals and fix them before they become serious problems.
- **Motion Detection** - Motion sensors can detect vibrations in buildings, bridges, dams and other large-scale structures. These devices can identify anomalies and disturbances in the structures that could lead to catastrophic failures. They can also be used in areas susceptible to floods, landslides, and earthquakes.

What is Sensor?

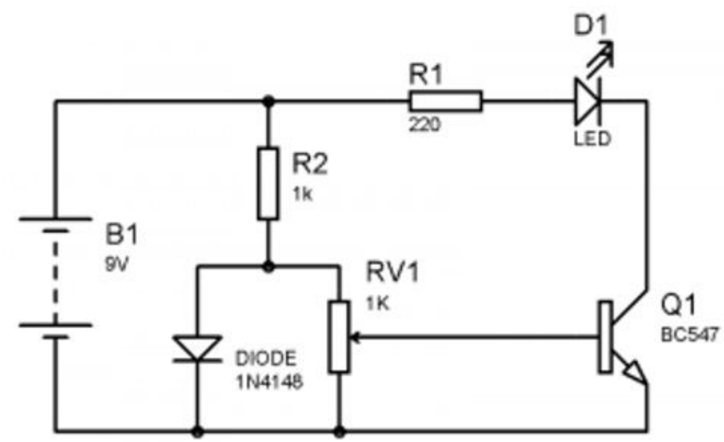
- A sensor is a device that produces an output signal for the purpose of sensing a physical phenomenon. In the broadest definition, a sensor is a device, module, machine, or subsystem that detects events or changes in its environment and sends the information to other electronics, frequently a computer processor.
- There are different types of sensors, for example, consider a thermocouple which can be considered as a temperature sensor that produces an output voltage based on the input temperature changes.

Sensors for IoT



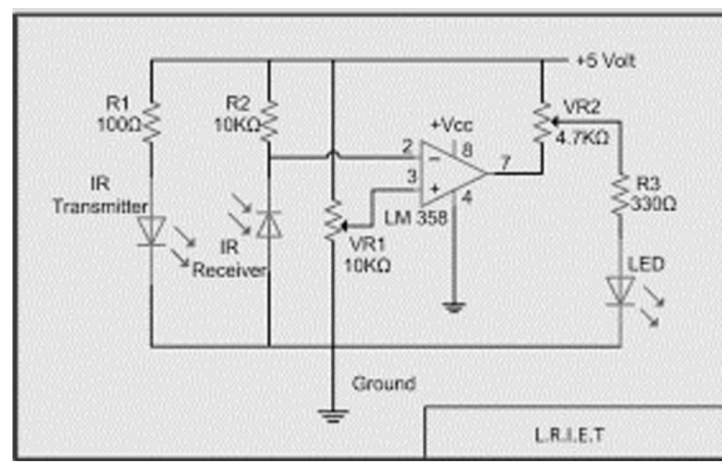
Temperature Sensor

- Temperature is one of the most commonly measured environmental quantities for different reasons.
- There are different types of temperature sensors that can measure temperature, such as a thermocouple, thermistors, semiconductor temperature sensors, resistance temperature detectors (RTDs), and so on.



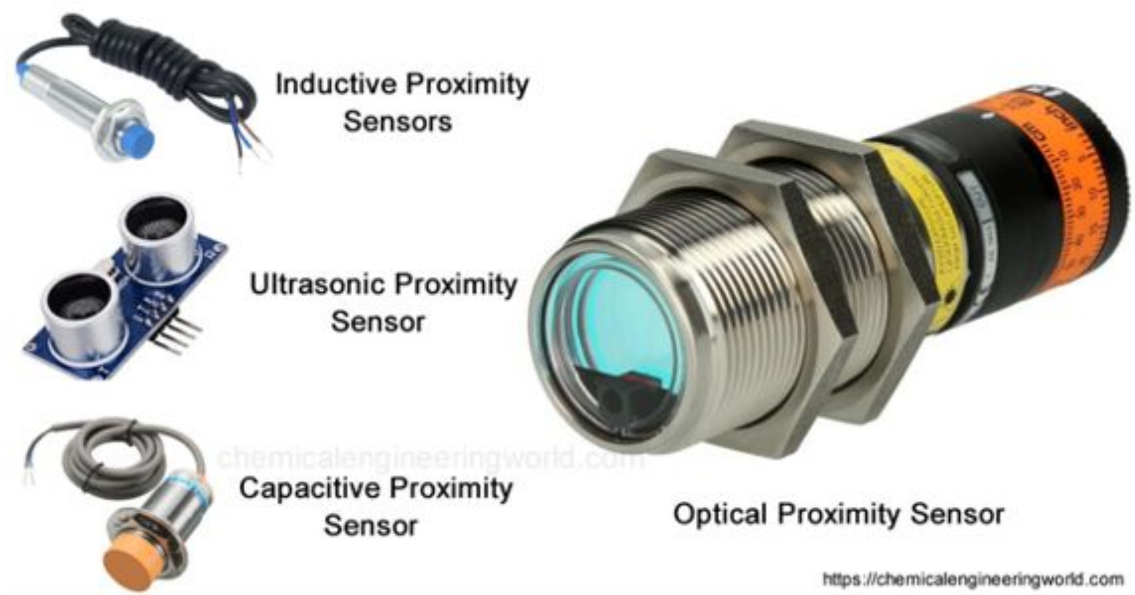
IR Sensor

- The small photo chips having a photocell which are used to emit and detect the infrared light are called as IR sensors.
- IR sensors are generally used for designing remote control technology. IR sensors can be used for detecting obstacles of the robotic vehicle and thus control the direction of the robotic vehicle.



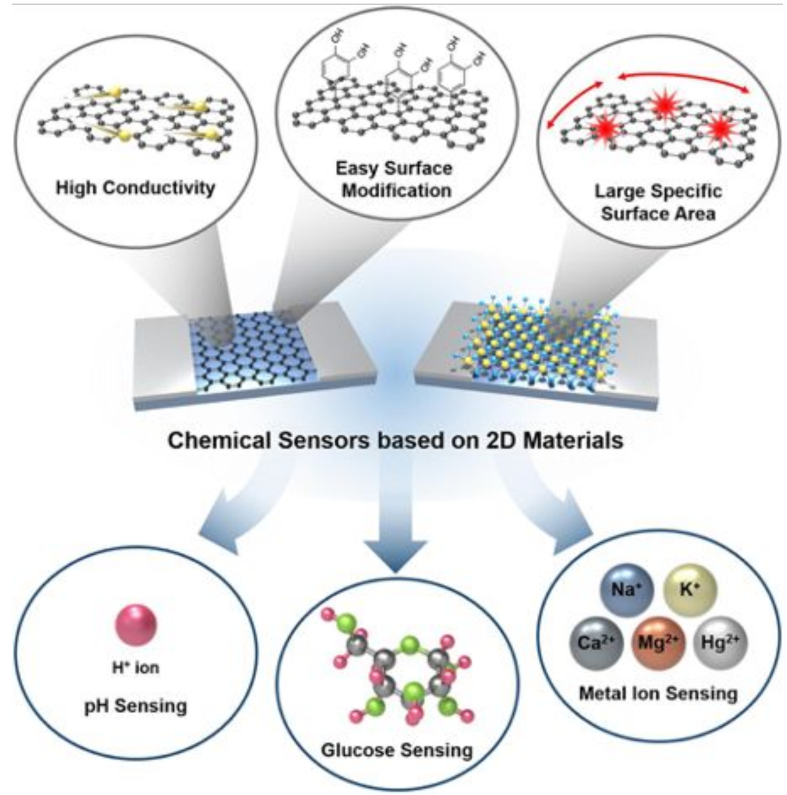
Proximity Sensors

- This a type of IoT sensor where it identifies the existence or non-existence of the surrounding object or finds the object properties.
- Then it converts the detected signal into the form that is clearly understood by the user or might be a simple electronic device that gets in no contact with them.



Chemical Sensors

- industries. The main objective of these sensors is to signify any kind of changes in the liquid or to detect any air chemical variations.
- These are crucially implemented in bigger towns and cities because it is important to look for changes and provide safety for the population.



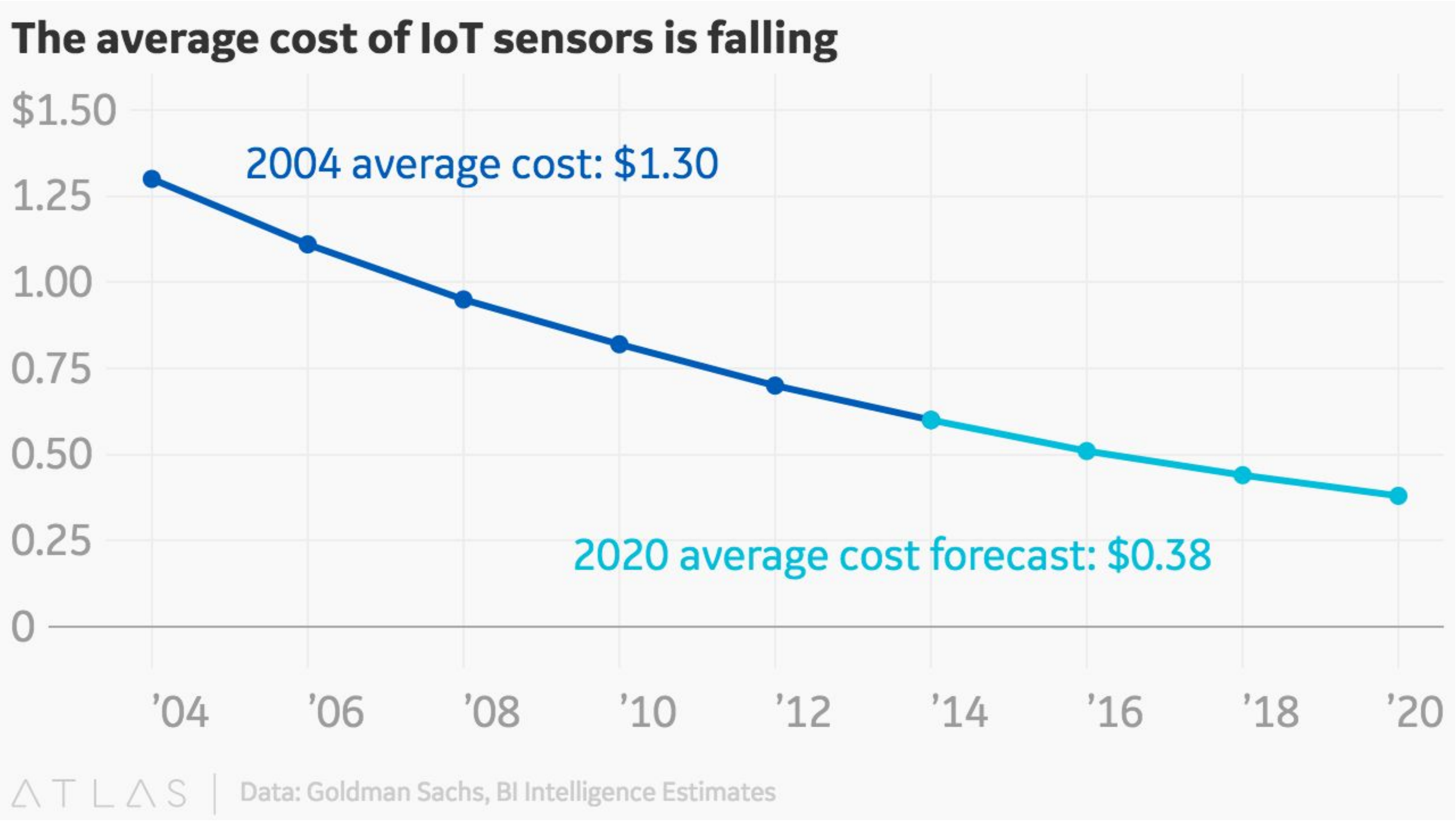
Gas Sensors

- These are almost the same as chemical sensors but are exclusively implemented to observe modifications of the quality of the air and to find out the existence of different types of gases.
- Examples
    - Hydrogen type
    - Ozone monitoring type
    - Hygrometer
    - Carbon-dioxide sensor
    - Electrochemical gaseous type
    - Catalytic bead type

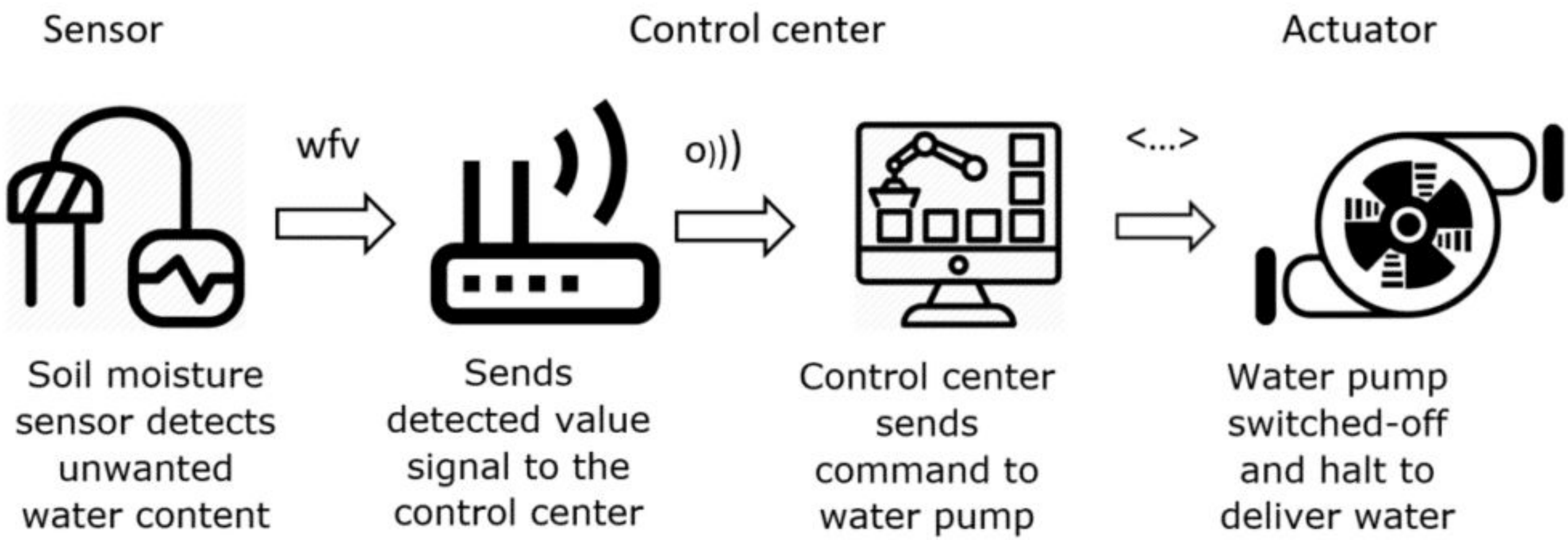
Humidity Sensors

- Humidity is the term that is specified as the amount of vapor that exists in the atmospheric air or in other gaseous substances.
- Humidity sensors generally adhere to the utilization of temperature sensors as because most of the manufacturing operations need exact operating conditions.

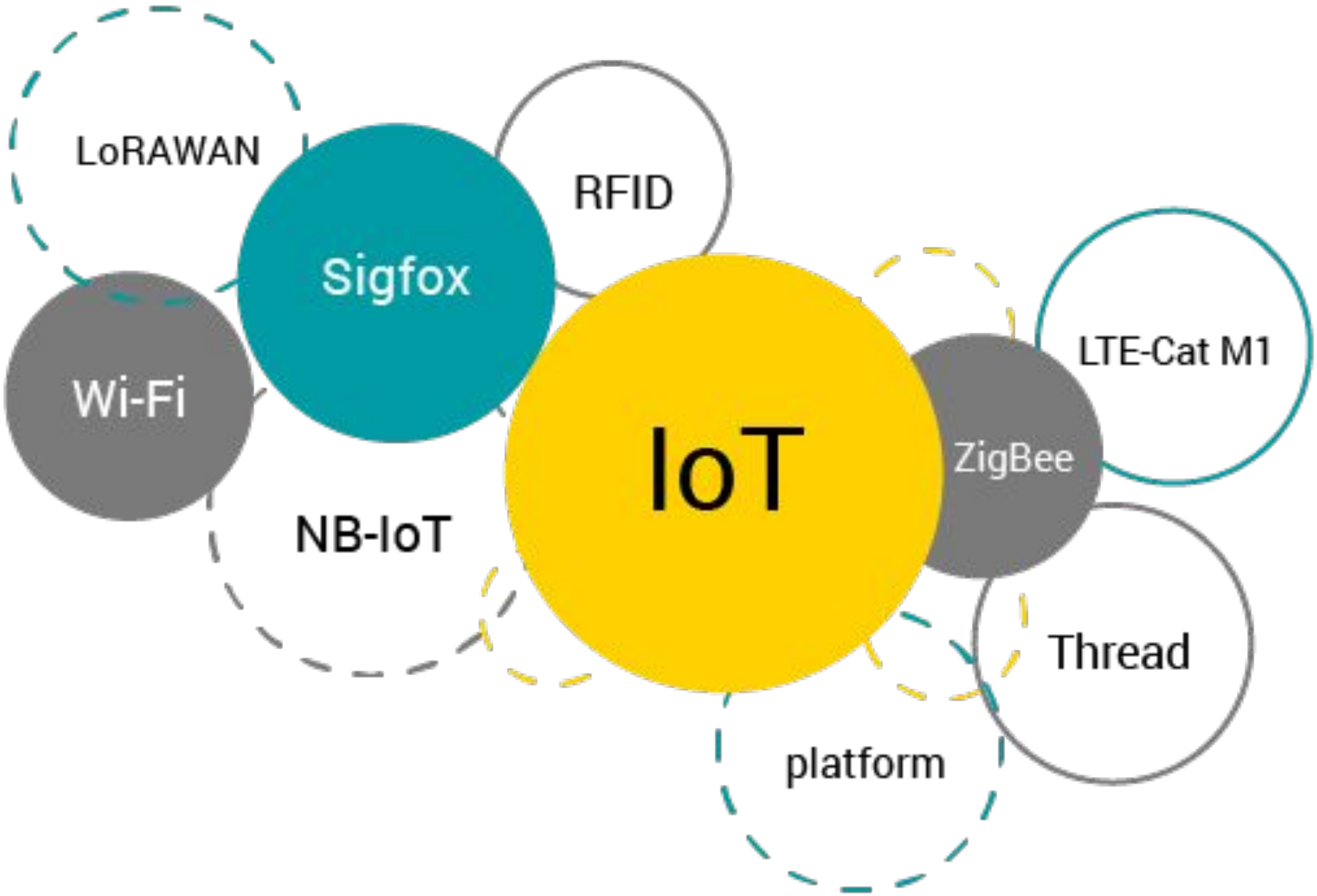
Sensors for IoT



IoT Actuator



Wireless Technologies for IoT



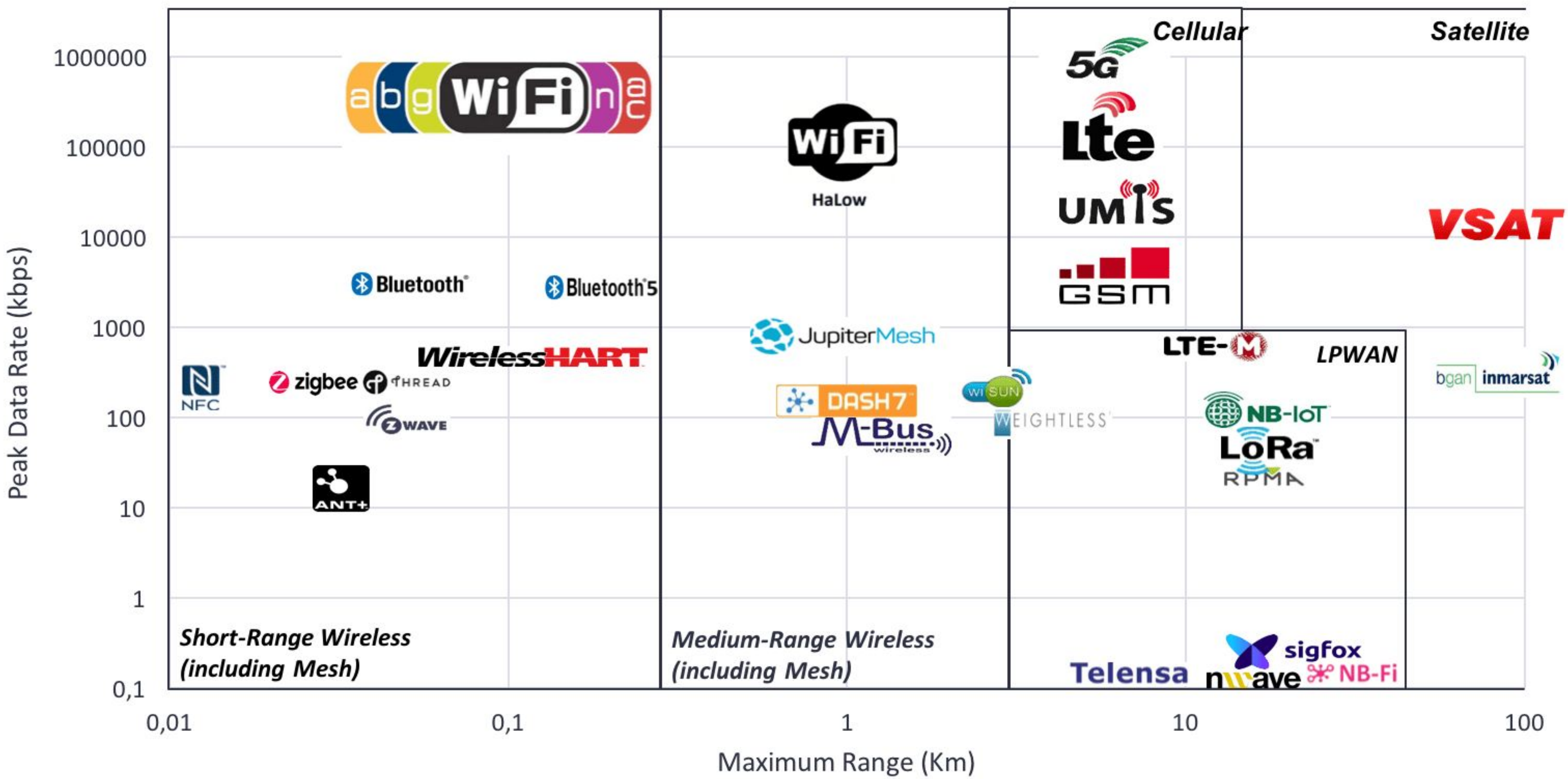
Wireless Technologies for IoT

- **Wi-Fi:** Wi-Fi is a widely-used wireless communication technology that offers high data rates and reliable connectivity over short to medium distances.
- **Bluetooth:** Bluetooth is a low-power, short-range wireless technology that is commonly used for connecting IoT devices to smartphones, tablets, and other devices.
- **Zigbee:** Zigbee is a low-power, wireless mesh networking technology that is designed for IoT applications that require low data rates, low power consumption, and reliable communication over longer distances.
- **Z-Wave:** Z-Wave is a wireless communication technology that is optimized for low-power IoT devices and offers reliable communication over longer distances.

Wireless Technologies for IoT

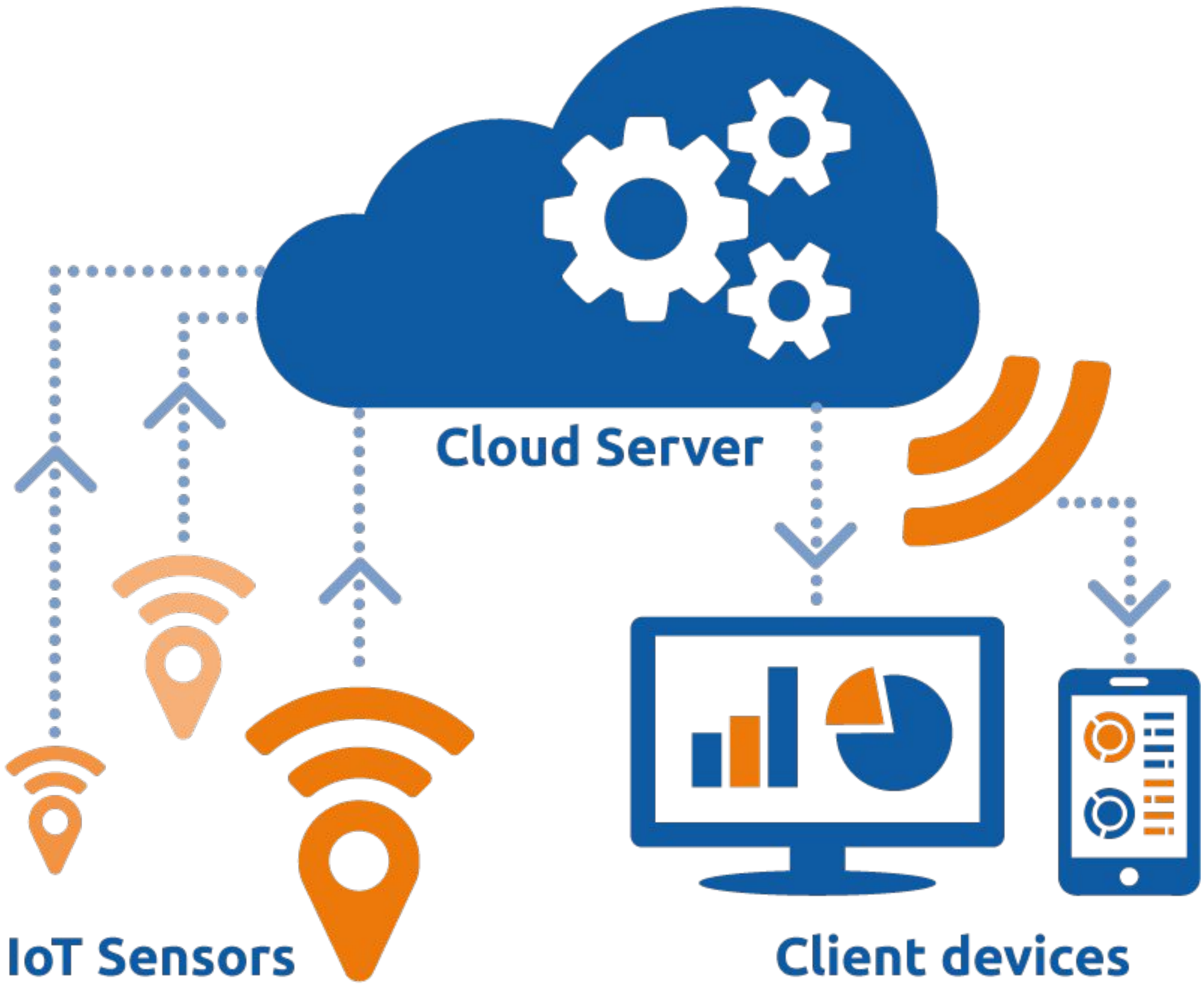
- **LoRaWAN:** LoRaWAN is a long-range, low-power wireless communication technology that is designed for IoT applications that require long-range communication, such as smart city applications.
- **Sigfox:** Sigfox is a low-power, wide-area network (LPWAN) technology that is designed for IoT applications that require low data rates and long-range communication, such as asset tracking and remote monitoring.
- **NB-IoT:** Narrowband IoT (NB-IoT) is a low-power, wide-area network (LPWAN) technology that is designed for IoT applications that require low data rates, long-range communication, and reliable connectivity.

Comparison of Wireless Technologies



Topic 2
Collect and Post Data to Cloud

Cloud Computing



Cloud Platforms

Google Cloud

amazon web services

Azure

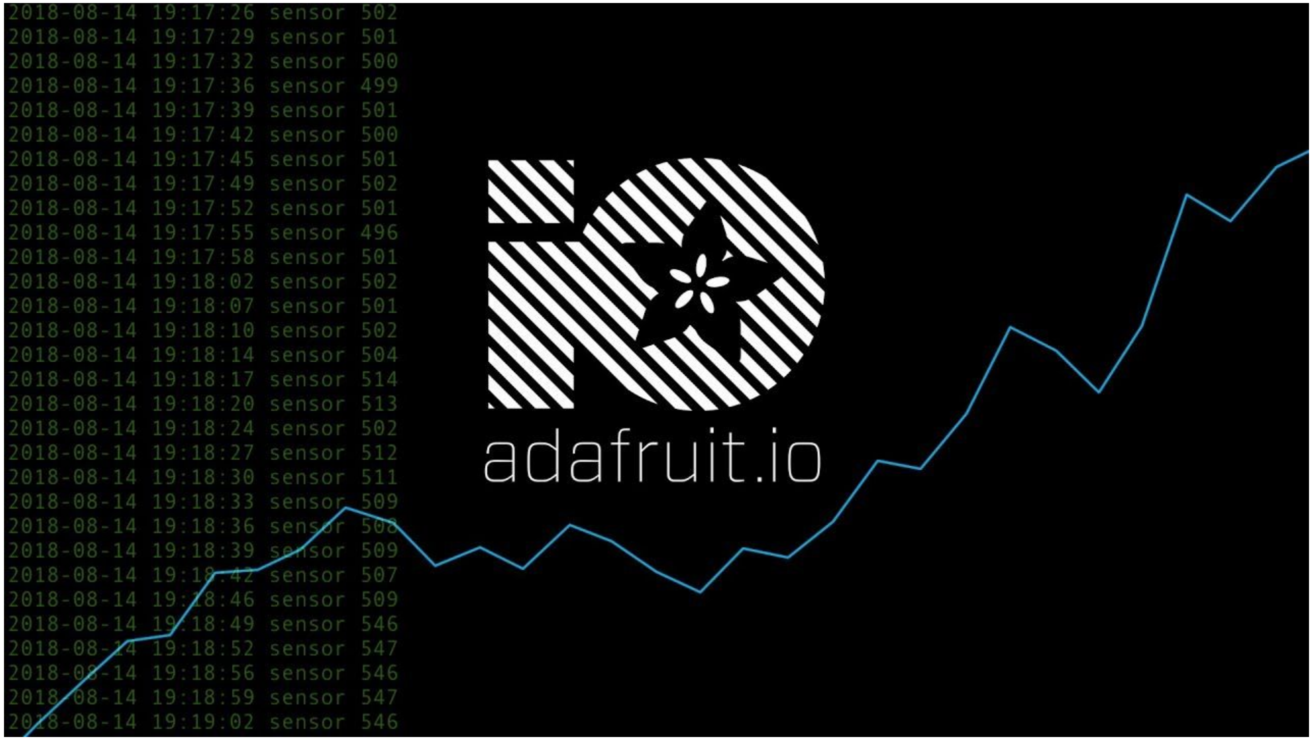
HUAWEI

Alibaba Cloud

ThingSpeak

adafruit

Adafruit.io



- Adafruit.io is a cloud service - where you can connect to it over the Internet, primarily for storing and retrieving data.
- Adafruit.io is designed to display, respond, and interact with your project's data. All data are keep private (data feeds are private by default) and secure.

Adafruit.io Features

Built from the ground up to be **easy to use.**

We do the hard work so you can focus on **the fun stuff.**

Powerful API

The same API that drives our user interface is available to you. We provide documentation so you can build a library in your favorite language to talk to IO, or use one of ours.

It's Your Data

The data you store with IO is yours to manage and control. You can download it all anytime and we will never sell or give it away to another company. We feel strongly enough about this that we put it in our IoT Bill of Rights.

Beautiful Dashboards

Data won't help you make better decisions or understand the world you're living in unless you can *see* it. Charts, graphs, gauges, logging, and more are available from anywhere in our carefully designed web-based dashboards.

Private & Secure

We won't share your account information with anyone without your permission and everything you store on Adafruit IO is **private by default**, whether you're a paying customer or not.

Documentation

We've written a lot more about Adafruit IO than just our web API. Teaching is at the heart of Adafruit, and our Learning System has all the projects and guides you need to get started or to level up!

Actions

Keep yourself and your systems up to date by automatically responding to changes. Actions monitor your data to notify you when your temperature sensor gets too hot or when your DIY security system goes offline.

Hardware

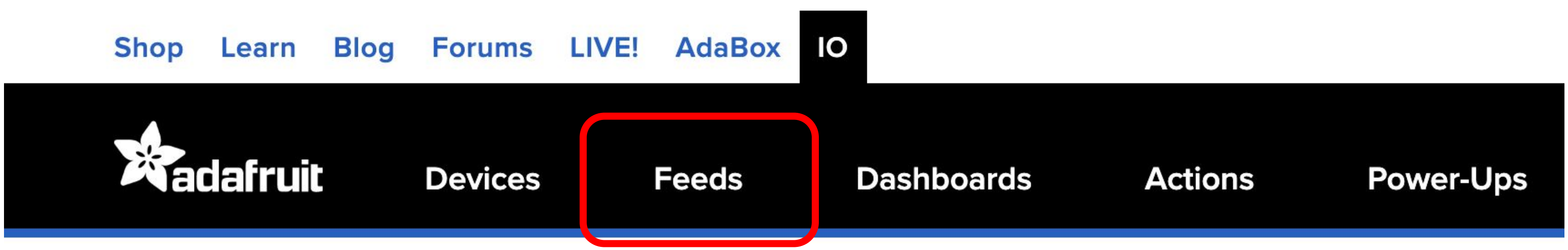
We have been building Internet of Things devices and kits since long before Adafruit IO ever existed. Start your project with the right equipment or pick up the one last thing you need to finish. We've got hardware for all types of makers.

Amazing Community

We Love Makers! Join us on the forums, on our Discord chat server, stop by the weekly Show and Tell live stream to share your project, and subscribe to our YouTube channel to see what we're up to.

Setup Adafruit.io Account

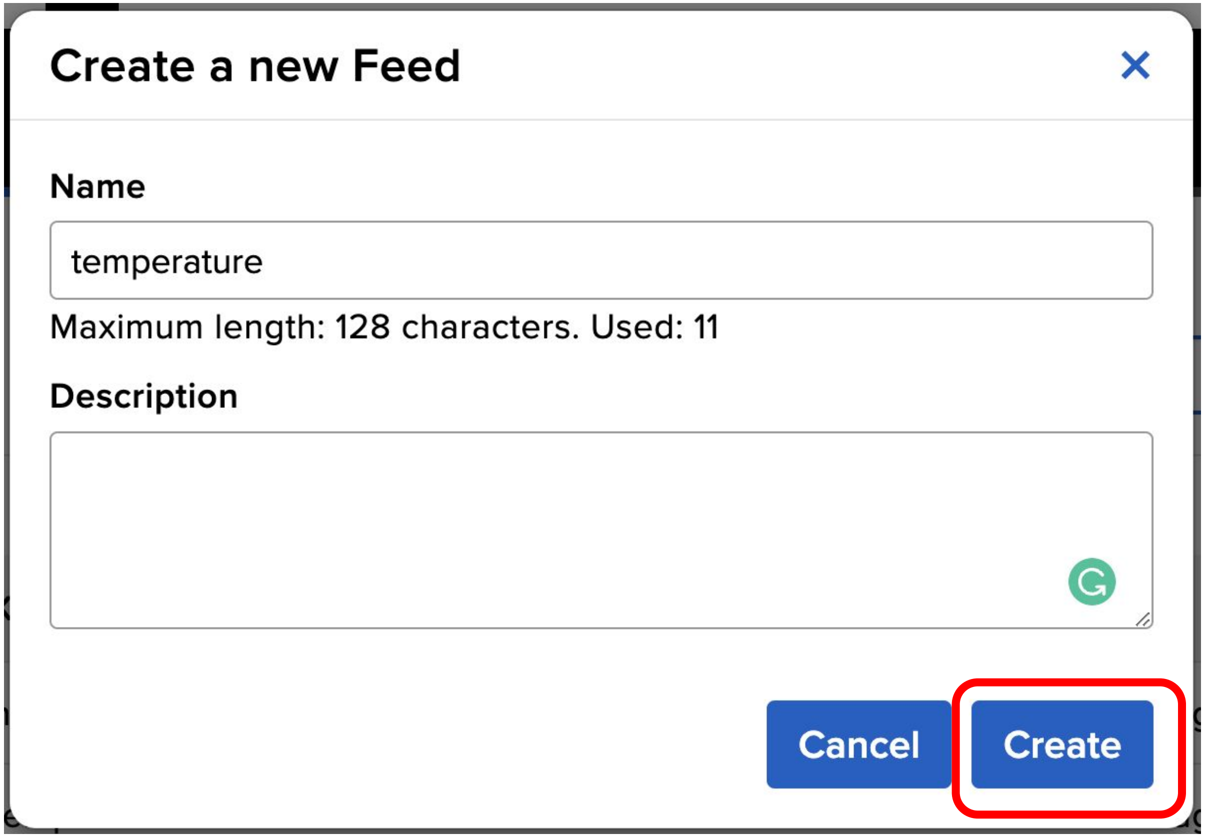
- Create a free Adafruit.io account or sign into Adafruit.io using an existing account (https://accounts.adafruit.com/users/sign_up)
- Once you've got your Adafruit.io account all set up, click the "IO" header and click "Feeds".
- Under the Feeds page, create a "New Feeds".
- The feeds are environmental data that are being sent from the DHT11 sensor.



- Remember your Adafruit.io username!

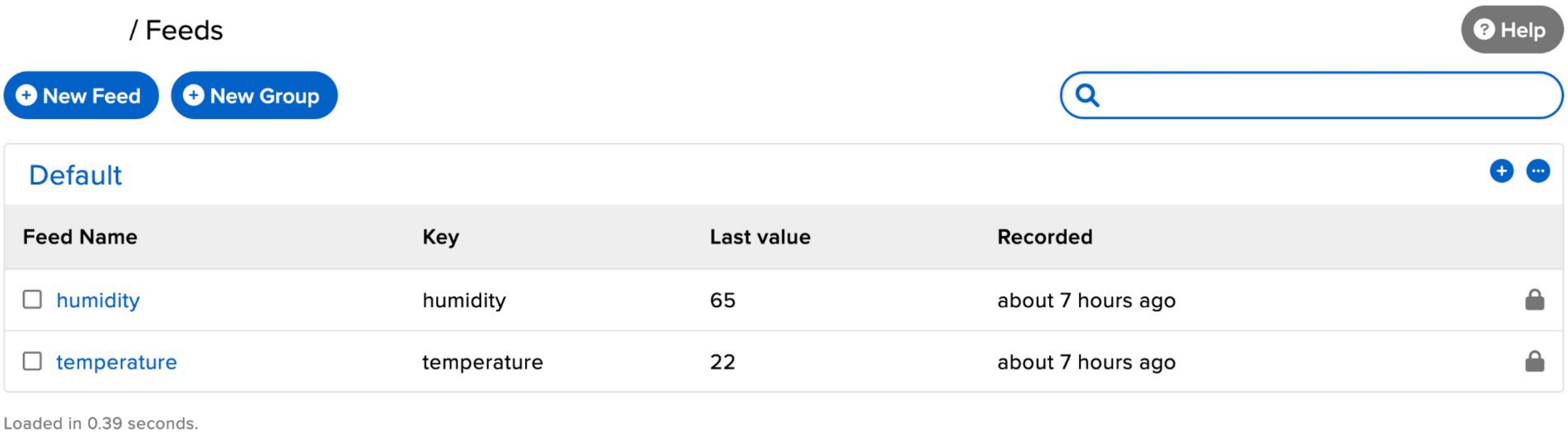
Create Adafruit.io Feeds

- After clicking on “New Feeds”, you will be prompted to give the new feed a name and description (optional).
- Once done, click the “Create” button.
- Repeat this step with another feed named “humidity”.



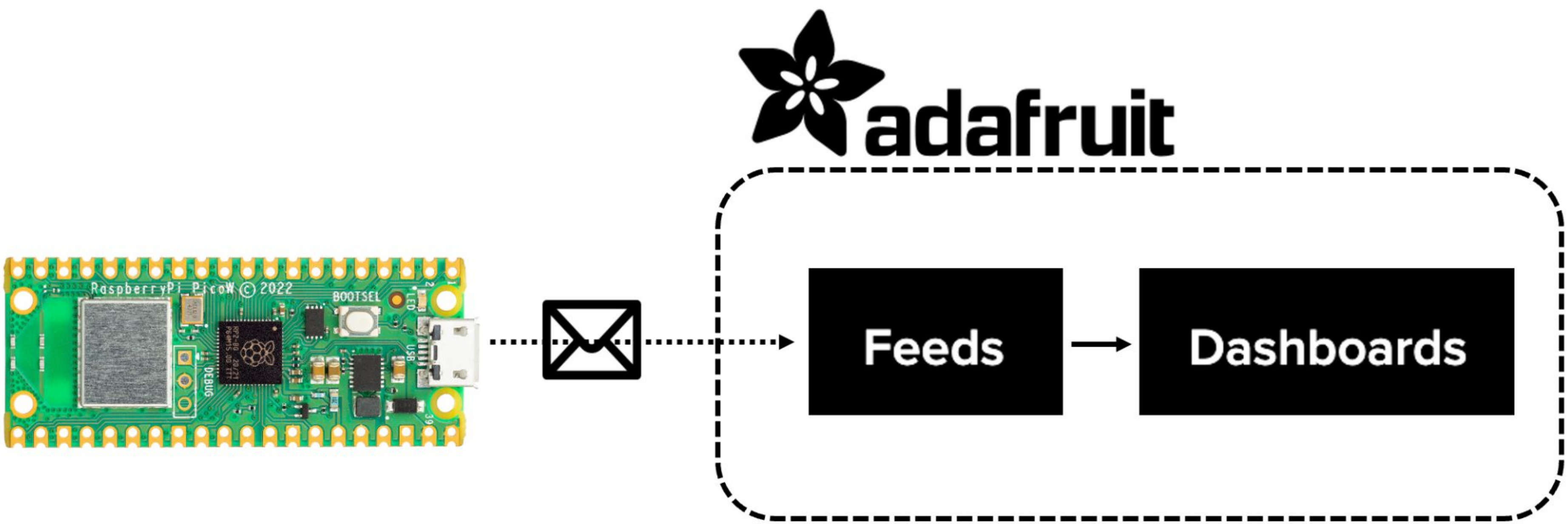
Completed Adafruit.io Feeds

- Once you have created two feeds, it should look something like this.



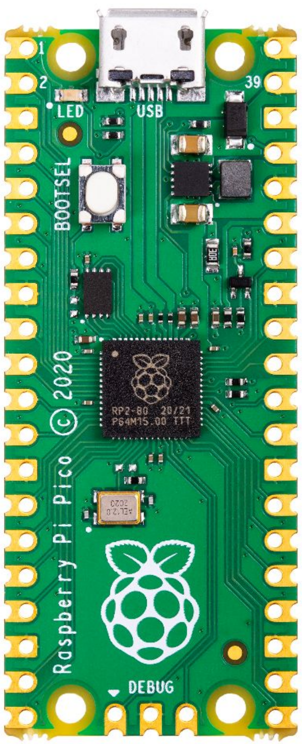
Raspberry Pi Pico W IoT

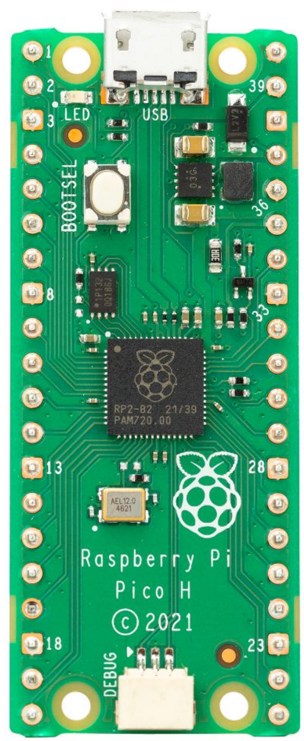
- To setup the Raspberry Pi Pico W IoT:
  - Collect data from sensors, eg DHT11 sensor
  - Upload the data to cloud, eg Adafruit.io feeds
  - Consume the incoming events to be visualized in real-time using Adafruit.io dashboard



Overview of Raspberry Pi Pico Series

- The Raspberry Pi Pico series is a low cost, high-performance microcontroller board.
- It enables people of all ages to explore computing, and to learn how to program in languages like C, C++ and MicroPython.

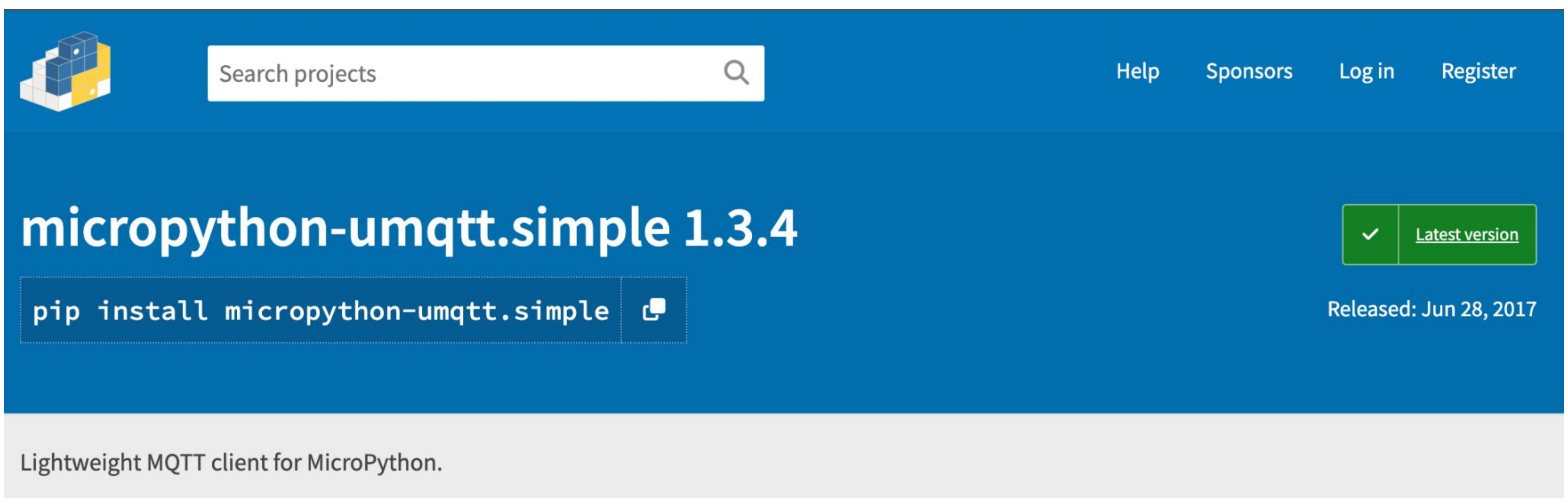




Overview of Raspberry Pi Pico W

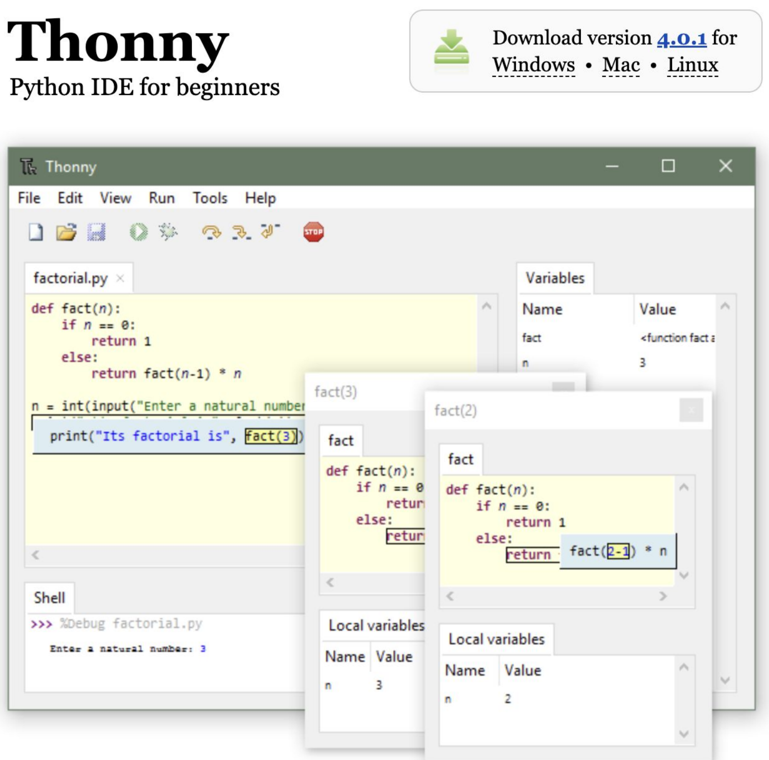
Installing a Python Package

- To allow us to communicate between the Raspberry Pi Pico W and the IoT Hub, we'll be using MQTT.
- To enable this, we're going to download a MicroPython IDE ("Thonny"), install the latest firmware and install a micropython module called "umqtt.simple" (https://pypi.org/project/micropython-umqtt.simple/).



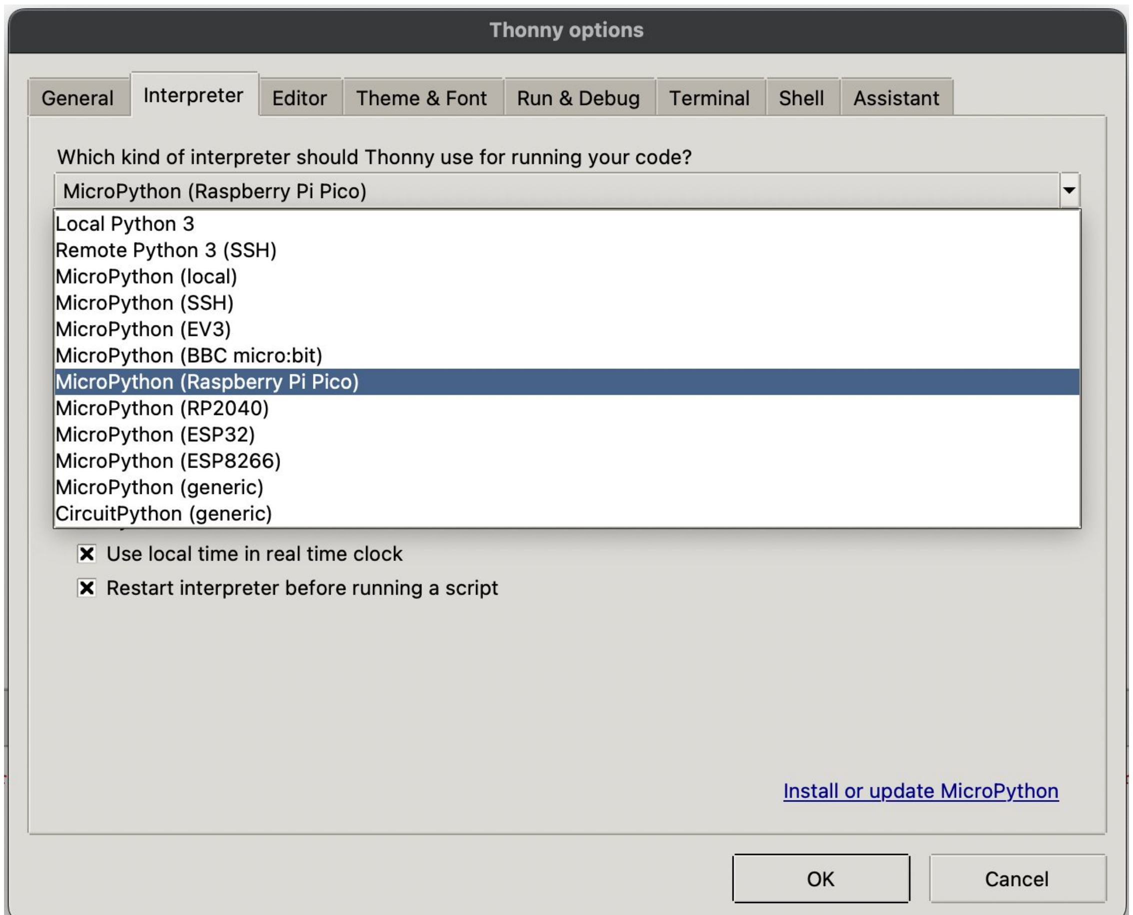
Download Thonny IDE

- Thonny IDE was part of the Raspberry Pi Operating System (formerly Raspbian) for ages.
- It is a Python IDE that is beginner friendly.
- Easy to get started and simple to debug codes.
- Download the Thonny IDE from https://thonny.org/



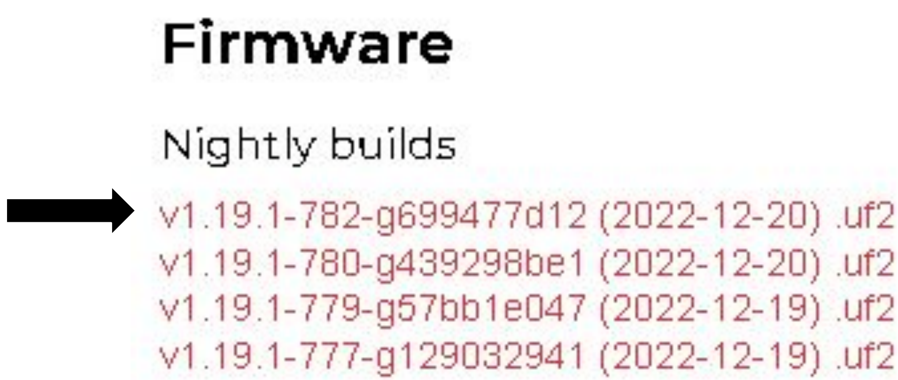
Configure Thonny IDE

- Open up Thonny IDE software.
- Go to '**Tools**' -> '**Options**', and under '**Interpreter**' tab, select '**MicroPython (Raspberry Pi Pico)**'.
- Click '**OK**' to close.



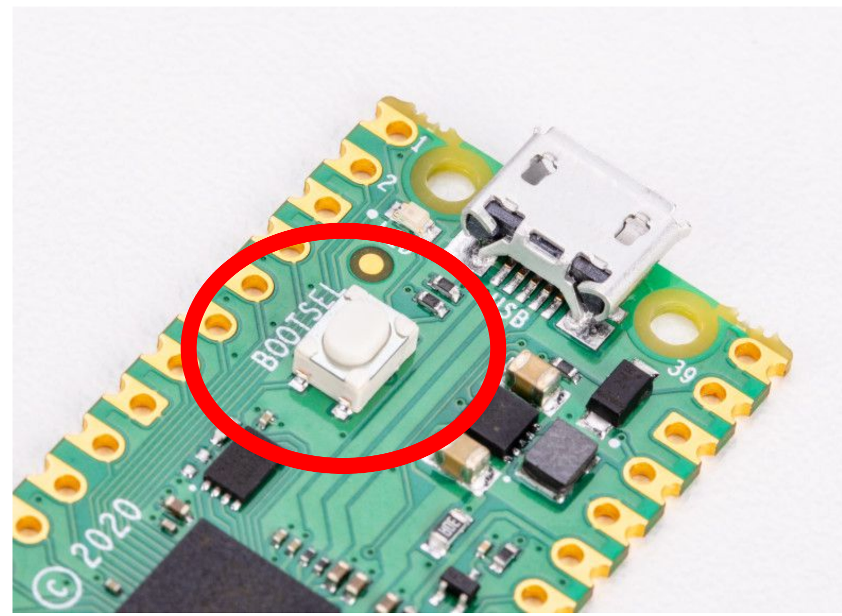
Download MicroPython UF2 file

- MicroPython is a lean and efficient implementation of the Python 3 programming that includes a small subset of the Python standard library.
- MicroPython aims to be as compatible with normal Python to allow code transfer from the desktop to a microcontroller with ease.
- Go to https://micropython.org/download/rp2-pico-w/
- Recommend to download the latest nightly build firmware.



Boot and Install MicroPython

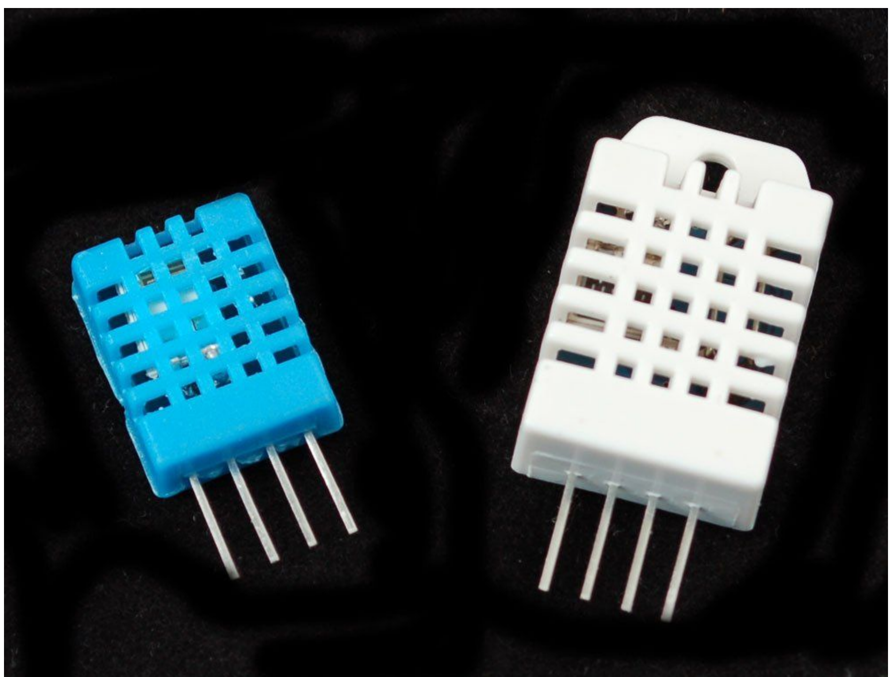
- To get started on the Raspberry Pi Pico W, you need to connect a microUSB cable to the Pico W board.
- Before you plug the other end into the computer, press and hold the BOOTSEL button.
- Continue to hold the button for a few more seconds after connecting to the computer before releasing.
- A new drive (RPI-RP2) will appear on your computer.
- Drag and drop the UF2 file onto the RPI-RP2 drive.



- The Pico W will auto reboot and run MicroPython.

DHT Sensors

- The DHT sensors are made of two parts, a capacitive humidity sensor and a thermistor.
- There is also a very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity.
- The two popular DHT sensors are DHT11 and DHT22.

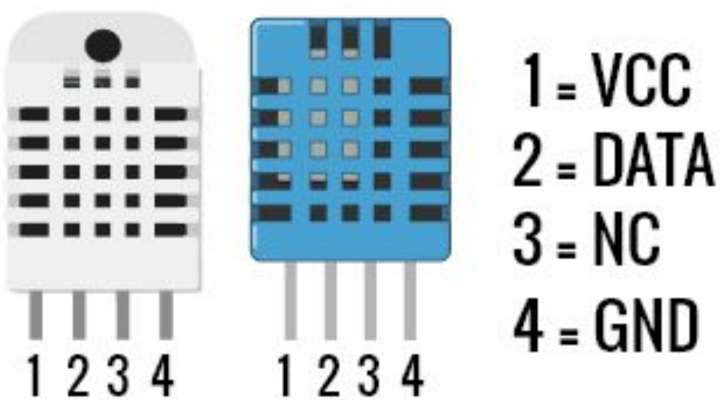


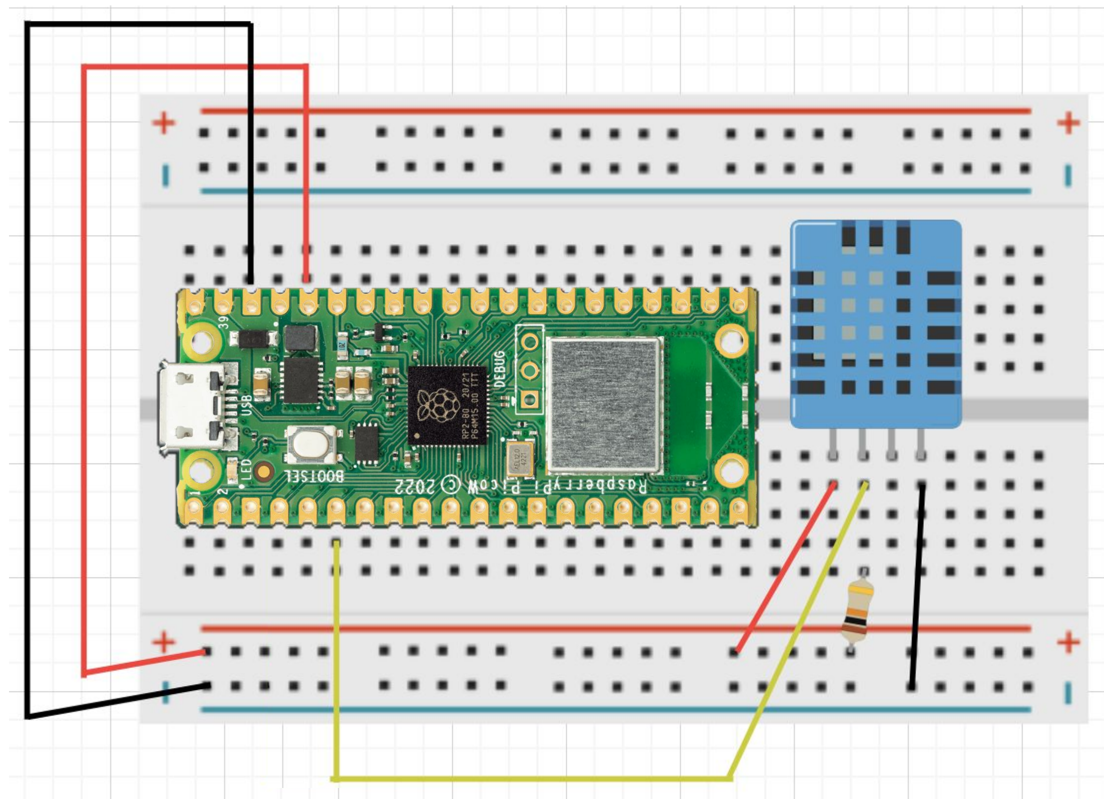
DHT11
- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

DHT22
- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1" spacing

Connect Pi Pico W to DHT Sensor

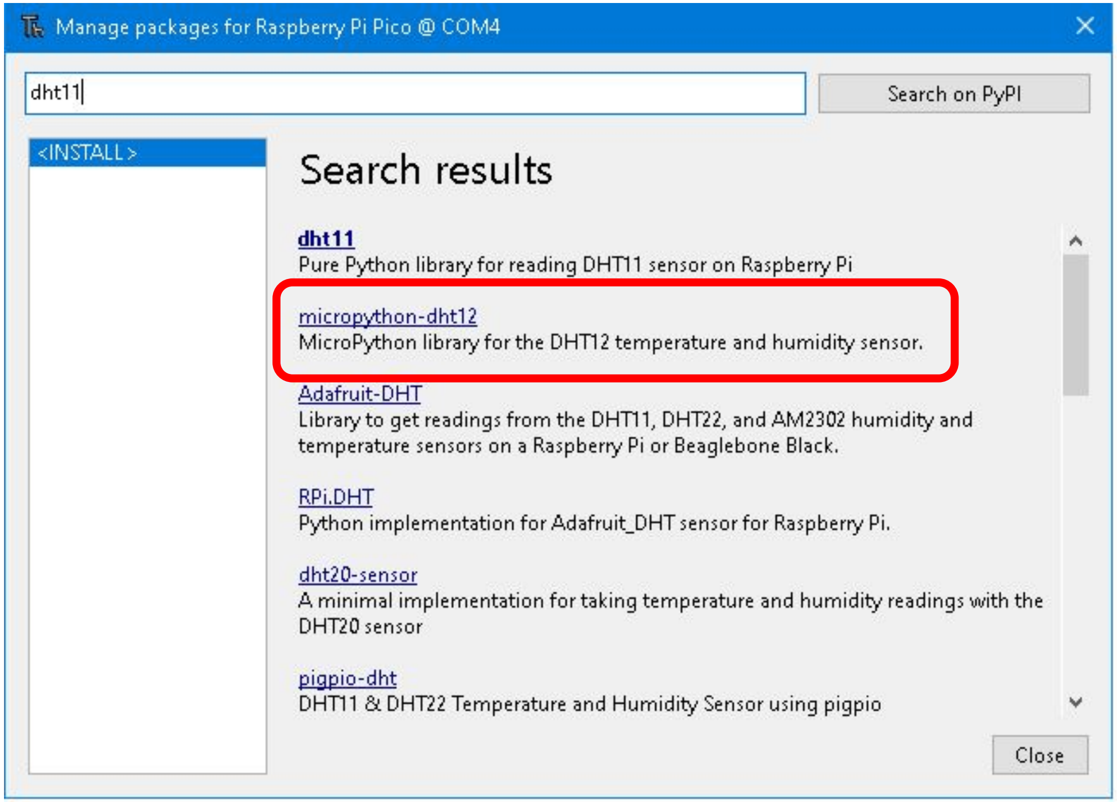
- Connect the DHT11 to Raspberry Pi Pico W as follows:
  - DHT Pin 1 to 3V3,
  - DHT Pin 2 to GP4,
  - DHT Pin 4 to GND.





Install DHT11 MicroPython Library

- In Thonny, click on **'Tools'** -> **'Manage packages…'** and type **'dht11'**.
- Select the micropython version and install it.



Thonny IDE

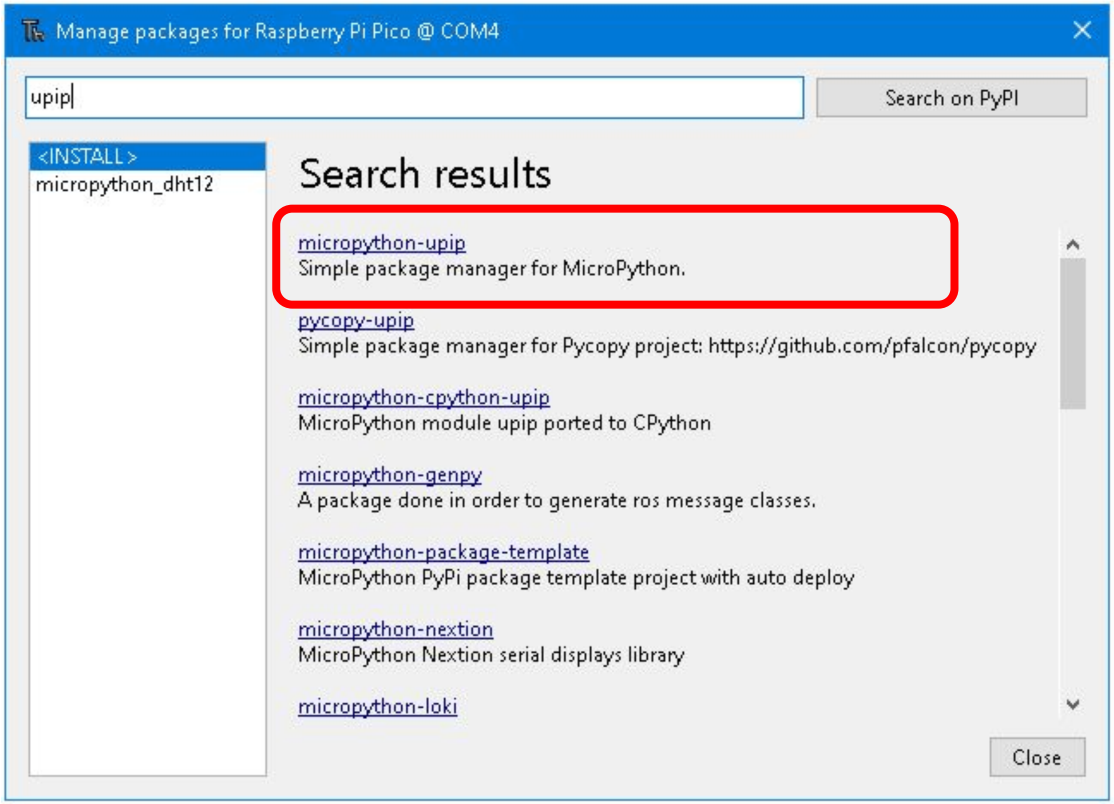
- Before proceeding, there are a few steps required.
- Open Thonny and in the Python shell, write the following lines to connect the Pi Pico W to Wi-Fi:

```
Shell
MicroPython v1.19.1-782-g699477d12 on 2022-12-20; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> import network
>>> wlan = network.WLAN(network.STA_IF)
>>> wlan.active(True)
>>> wlan.connect('Wi-Fi AP', 'PASSWORD')
```

- [Important !!!] Change the **"Wi-Fi AP"** and **"PASSWORD"** to match the details of your Wi-Fi Access Point.
- Remember to press Enter at the end of each line.

Install upip MicroPython Library

- In Thonny, click on '**Tools**' -> '**Manage packages...**' and type '**upip**'.
- Select the micropython version and install it.



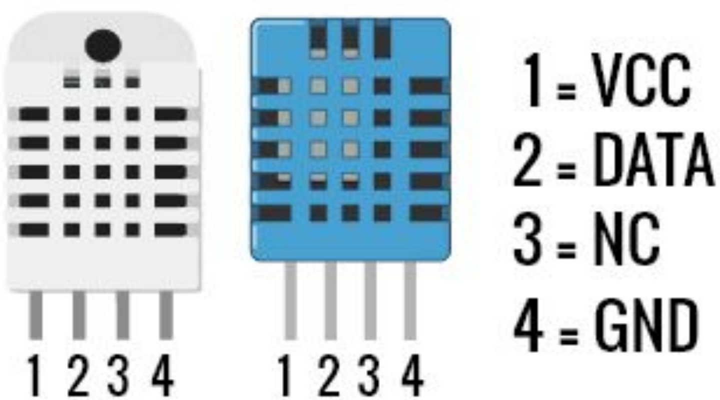
Thonny IDE

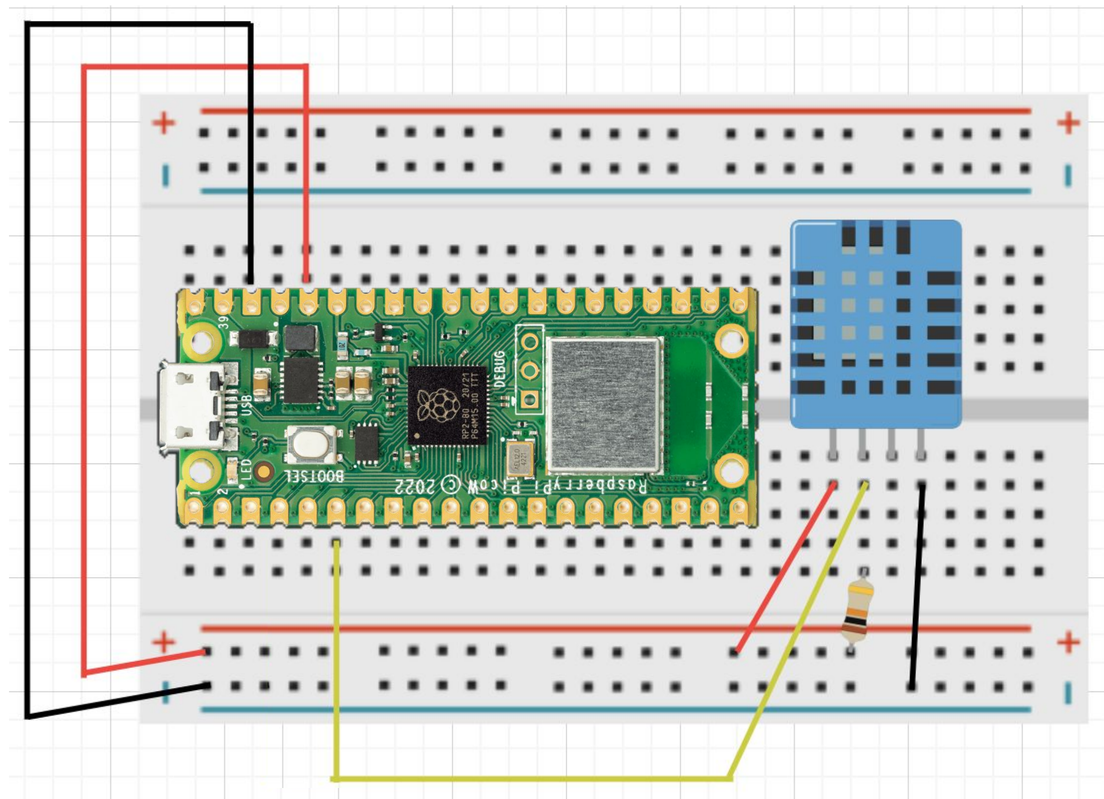
- Import the upip module. The upip module is a version of the Python package manager, pip, for MicroPython.
- Using upip, install ‘micropython-umqtt.simple’ in the ‘/lib’ directory. This module is a simplified MQTT client for MicroPython.

```
Shell

MicroPython v1.19.1-782-g699477d12 on 2022-12-20; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> import network
>>> wlan = network.WLAN(network.STA_IF)
>>> wlan.active(True)
>>> wlan.connect('Wi-Fi AP', 'PASSWORD')
>>> import upip
>>> upip.install('micropython-umqtt.simple', '/lib')

  Installing to: /lib/
  Warning: pypi.org SSL certificate is not validated
  Installing micropython-umqtt.simple 1.3.4 from https://files.pythonhosted.org/packages/bd/cf
  /697e3418b2f44222b3e848078b1e33ee76aedca9b6c2430ca1b1aec1ce1d/micropython-umqtt.simple-1.3.4
  .tar.gz
```

Activity: Setup Raspberry Pi Pico W To Sense Environmental Data

- Setup the Raspberry Pi Pico W to gather environmental data using the DHT11 sensor.





Sensing Environmental Data

```
from machine import Pin
from time import sleep
import dht

sensor = dht.DHT11(Pin(4))

while True:
    sensor.measure()
    temp = sensor.temperature()
    hum = sensor.humidity()
    print(“Temperature: {}°C Humidity: {}% ”. format(temp, hum))
    sleep(2)
```

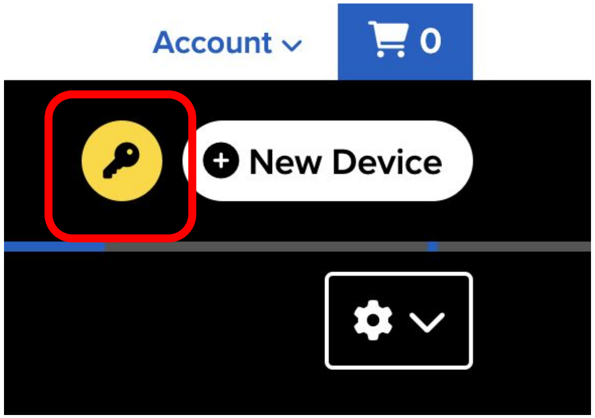
MicroPython DHT11 Pico W Code

- The MicroPython script reads temperature and humidity readings from the DHT11 sensor.
- These readings will be printed on the MicroPython shell console as seen below.

```
Shell
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
 Temperature: 23.8°C   Humidity: 58%
 Temperature: 23.5°C   Humidity: 58%
 Temperature: 23.5°C   Humidity: 58%
 Temperature: 23.5°C   Humidity: 58%
 Temperature: 23.4°C   Humidity: 58%
 Temperature: 23.4°C   Humidity: 58%
 Temperature: 23.4°C   Humidity: 58%
 Temperature: 23.4°C   Humidity: 58%
```

Sending Data To The Cloud

- There are a few things to note before you continue to program the Micropython code.
- Change the WIFI_SSID & WIFI_PASSWORD in order to connect to the Raspberry Pi Pico W to the network.
- You will need your unique Adafruit IO Key, mainly when sending data to the cloud, which can be located as shown below.



Sending Data To The Cloud

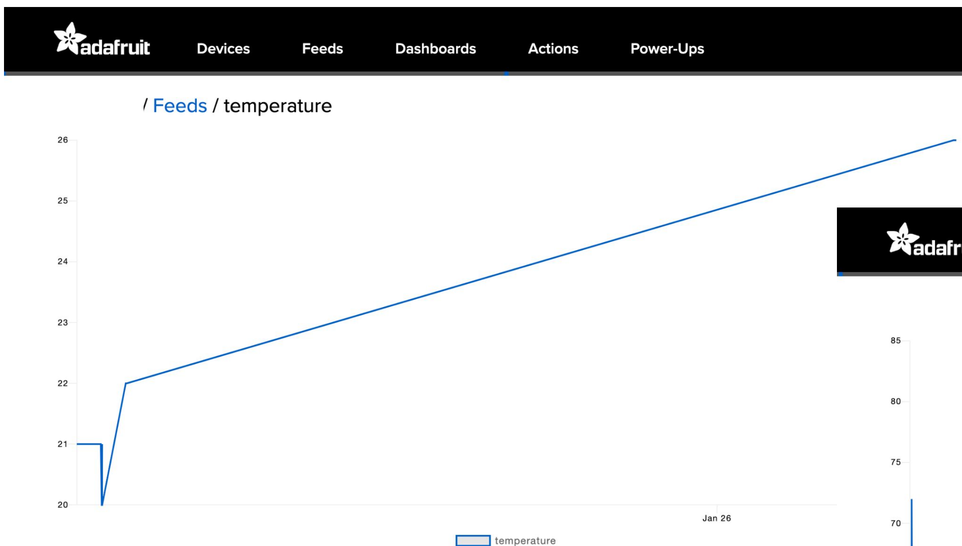
```
from machine import Pin
from time import sleep
from machine import Timer
from umqtt.simple import MQTTClient
import dht
import network
import time
import sys

sensor = dht.DHT11(Pin(4))

WIFI_SSID     = 'Wi-Fi AP'
WIFI_PASSWORD = 'PASSWORD'

mqtt_client_id      = bytes('client_'+'12321', 'utf-8') # Just a random client ID

ADAFRUIT_IO_URL     = 'io.adafruit.com'
ADAFRUIT_USERNAME   = 'XXXXXXXX'
ADAFRUIT_IO_KEY     = 'XXXXXXXX'

TEMP_FEED_ID      = 'temperature'
HUM_FEED_ID       = 'humidity'

def connect_wifi():
    wifi = network.WLAN(network.STA_IF)
    wifi.active(True)
    wifi.disconnect()
    wifi.connect(WIFI_SSID,WIFI_PASSWORD)
    if not wifi.isconnected():
        print('Connecting..')
        timeout = 0
        while (not wifi.isconnected() and timeout < 5):
            print(5 - timeout)
            timeout = timeout + 1
            time.sleep(1)
    if(wifi.isconnected()):
        print('Connected!')
    else:
        print('Not connected!')
```

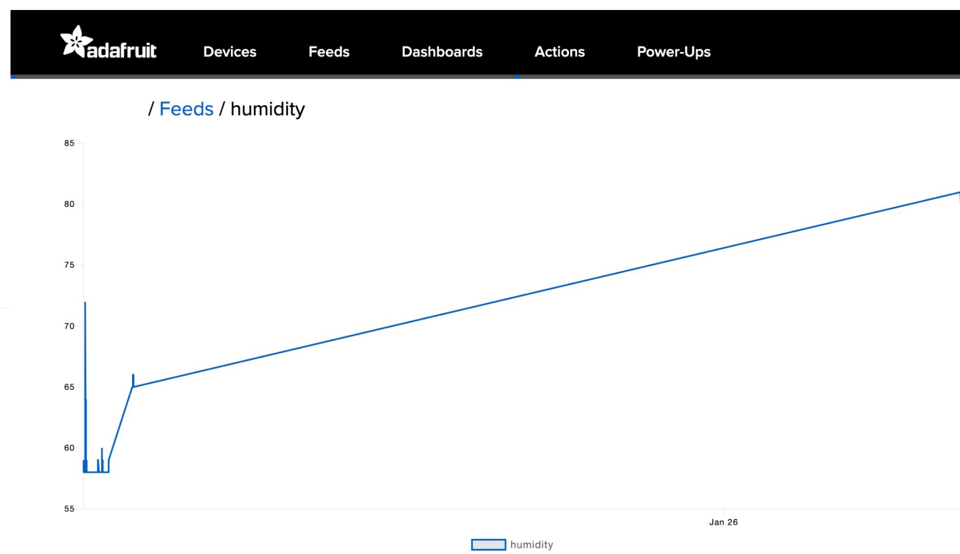
Sending Data To The Cloud

```
        print('Not connected!')
        sys.exit()

connect_wifi() # Connecting to WiFi Router

client = MQTTClient(client_id=mqtt_client_id,
                    server=ADAFRUIT_IO_URL,
                    user=ADAFRUIT_USERNAME,
                    password=ADAFRUIT_IO_KEY,
                    ssl=False)
try:
    client.connect()
except Exception as e:
    print('Could not connect to MQTT server {}{}'.format(type(e).__name__, e))
    sys.exit()

temp_feed = bytes('{:s}/feeds/{:s}'.format(ADAFRUIT_USERNAME, TEMP_FEED_ID), 'utf-8') # format - techiesms/feeds/temp
hum_feed = bytes('{:s}/feeds/{:s}'.format(ADAFRUIT_USERNAME, HUM_FEED_ID), 'utf-8') # format - techiesms/feeds/hum

def sens_data(data):
    sensor.measure()                   # Measuring
    temp = sensor.temperature()        # getting Temp
    hum = sensor.humidity()
    client.publish(temp_feed,
                  bytes(str(temp), 'utf-8'),   # Publishing Temp feed to adafruit.io
                  qos=0)

    client.publish(hum_feed,
                  bytes(str(hum), 'utf-8'),   # Publishing Hum feed to adafruit.io
                  qos=0)
    print("Temperature : ", str(temp))
    print("Humidity : " , str(hum))
    print(' ')

timer = Timer(-1)
timer.init(period=5000, mode=Timer.PERIODIC, callback = sens_data)
```

Adafruit.io DHT11 Pico W Code

- The MicroPython script reads temperature and humidity readings from the DHT11 sensor.
- These readings will be sent to the Adafruit.io individual feeds as seen below.





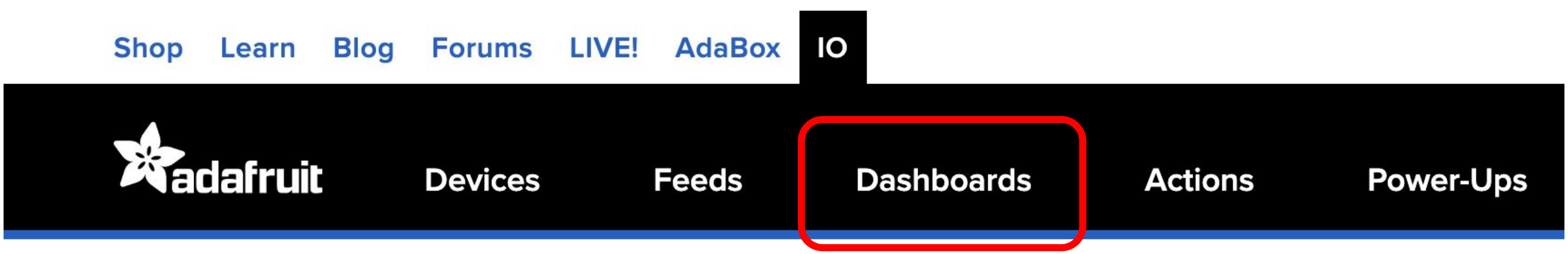
Topic 3
IoT Data Analytics and Visualization

IoT Data Analytics

- We have successfully sent environmental data to the cloud (respective Adafruit.io feeds) that we have created earlier.
- Now, we will convert these data and display them into a dashboard to better visualize the environmental data.
- In order to do so, we will need to create a Adafruit.io Dashboard first.

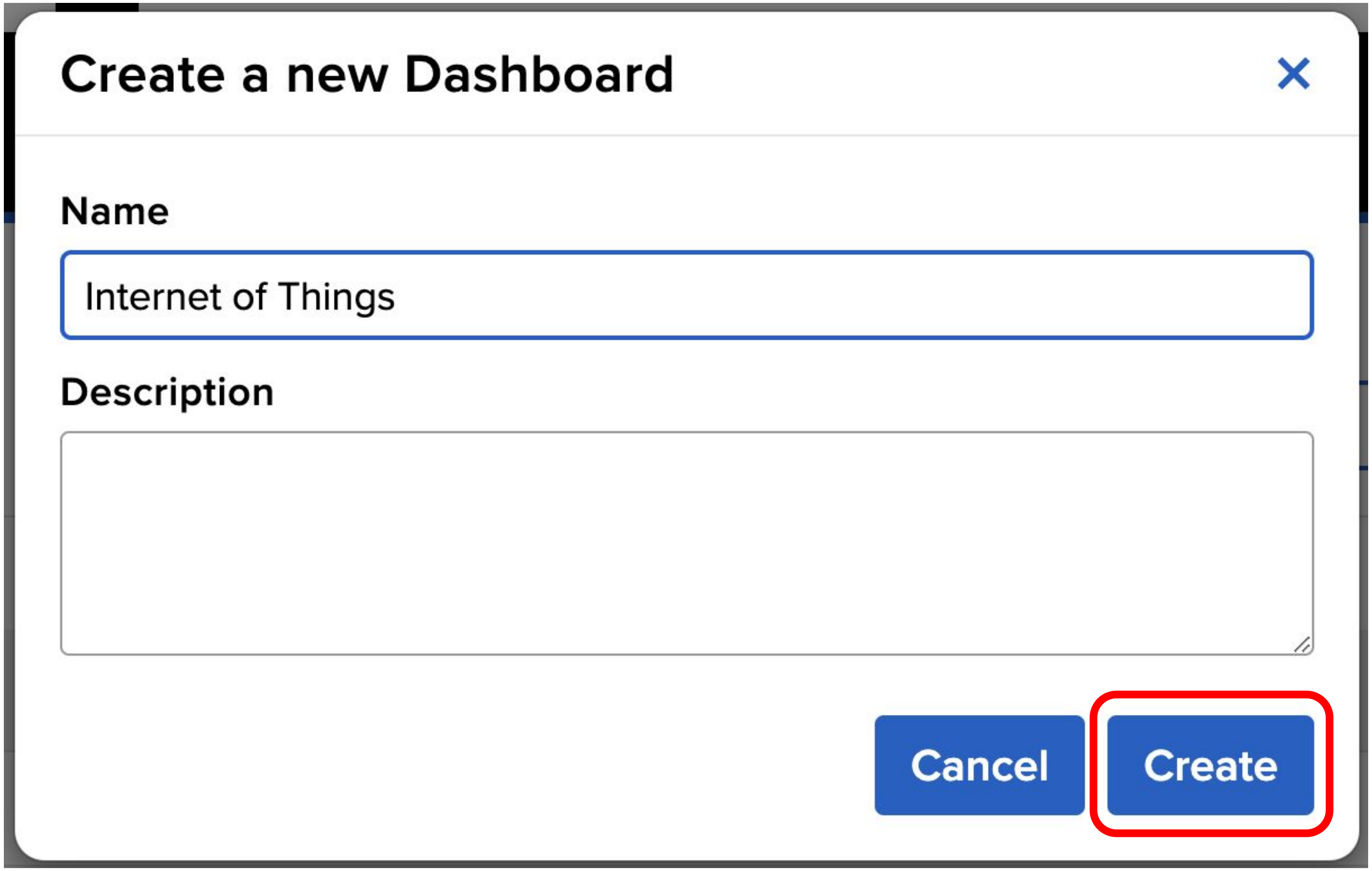
Create Adafruit.io Dashboard

- We have created two feeds (‘temperature’ & ‘humidity’) that are receiving environmental data.
- Under the “IO” header, click “Dashboard” and create a “New Dashboard”.
- The dashboard will be used to display the environmental data that are recorded in the feeds.



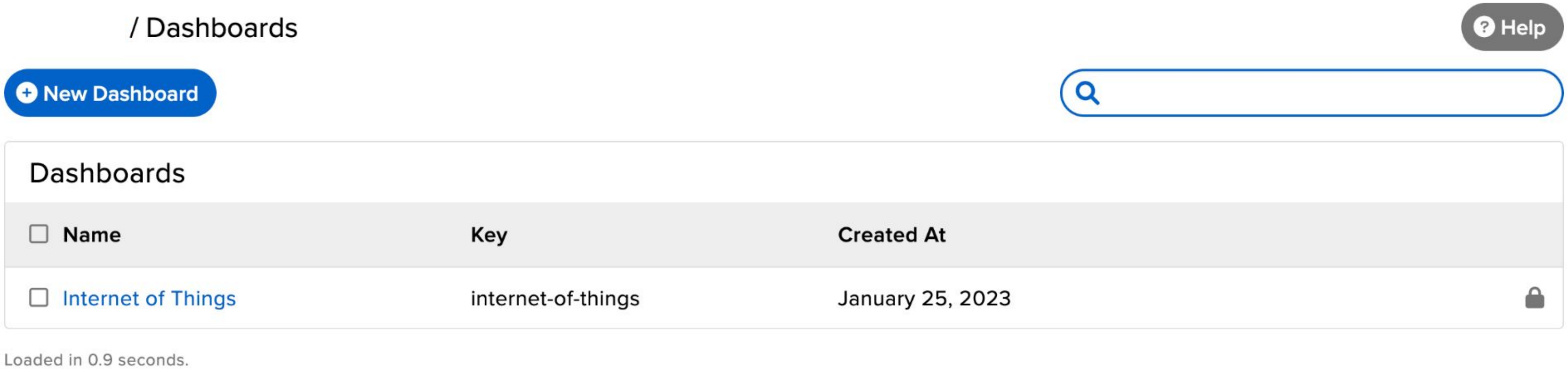
Create Adafruit.io Dashboard

- After clicking on “New Dashboard”, you will be prompted to give the dashboard a name and description (optional).
- Once done, click the “Create” button.



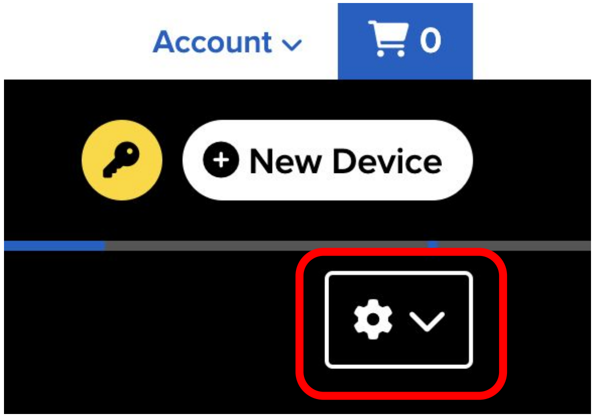
Completed Adafruit.io Dashboard

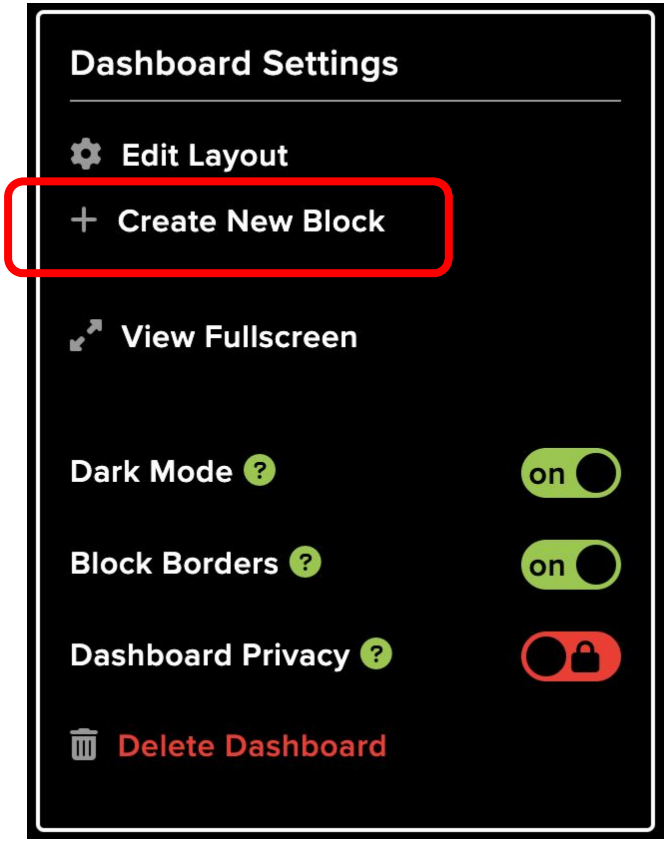
- Once you have created the dashboard, it should look like this.
- Click on the newly created dashboard ('Internet of Things').



Setup Dashboard Blocks

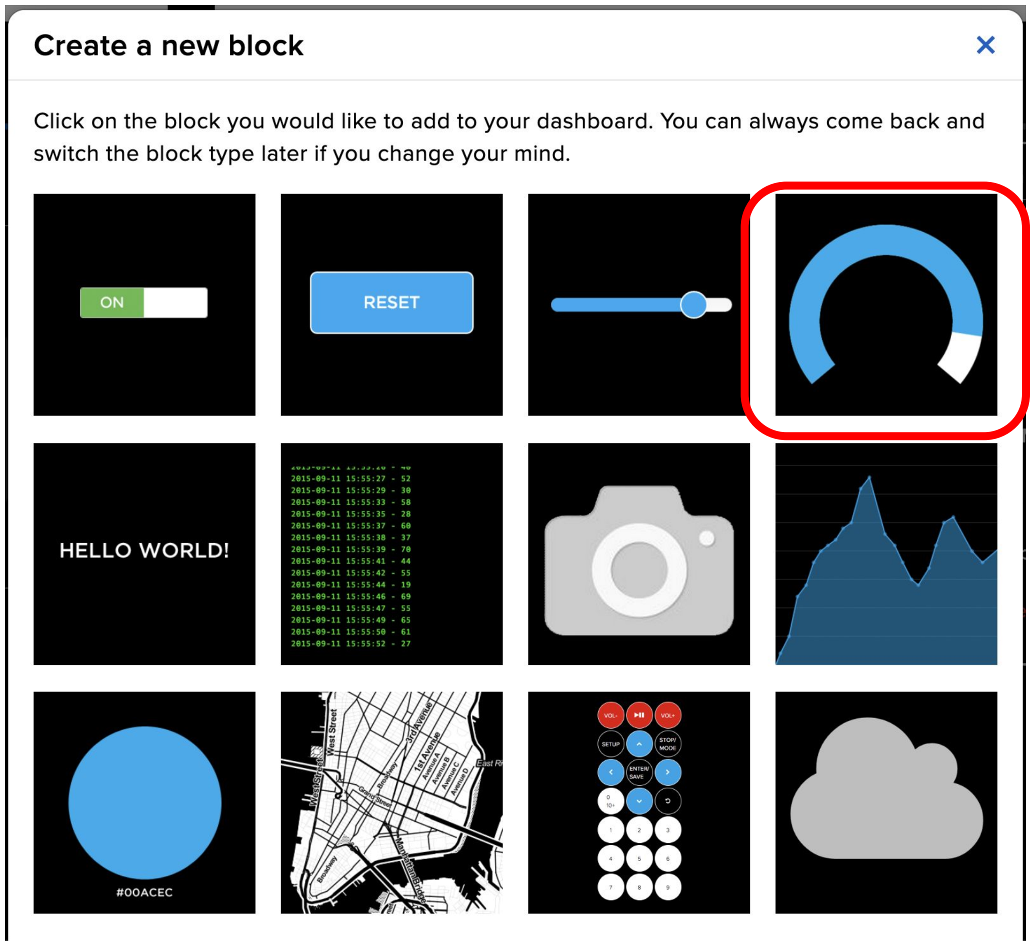
- On the newly created dashboard, you will not see any blocks.
- We will create two blocks that will be used to display the environmental data on the dashboard.
- Click on the settings dropdown icon and click “Create New Block”.





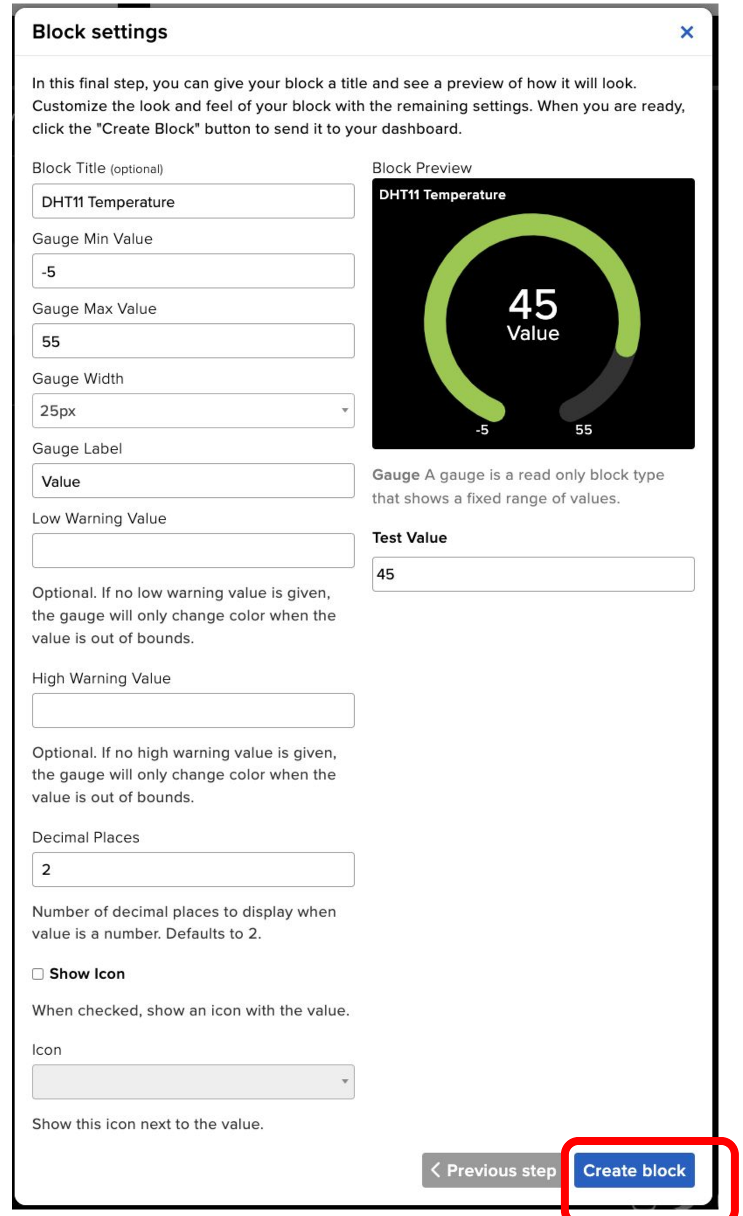
Setup Dashboard Blocks

- You will have many options to choose from.
- For now, choose the gauge block.
- Select the 'temperature' feed and click "Next step".



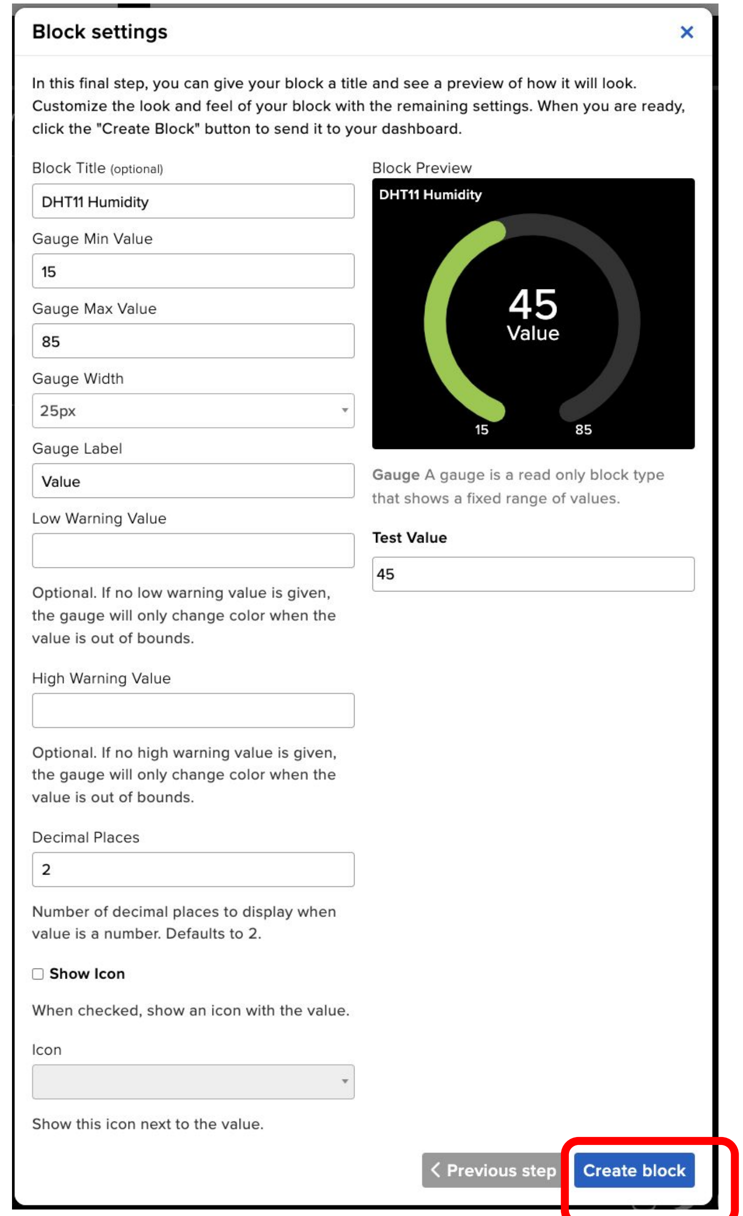
Setup Temperature Block

- Under the Block settings, setup the block as shown for the Temperature gauge:
  - Block Title (optional)
  - Gauge Min Value
  - Gauge Max Value
- Once done, click “Create Block”.
- Repeat the same process for the Humidity gauge.



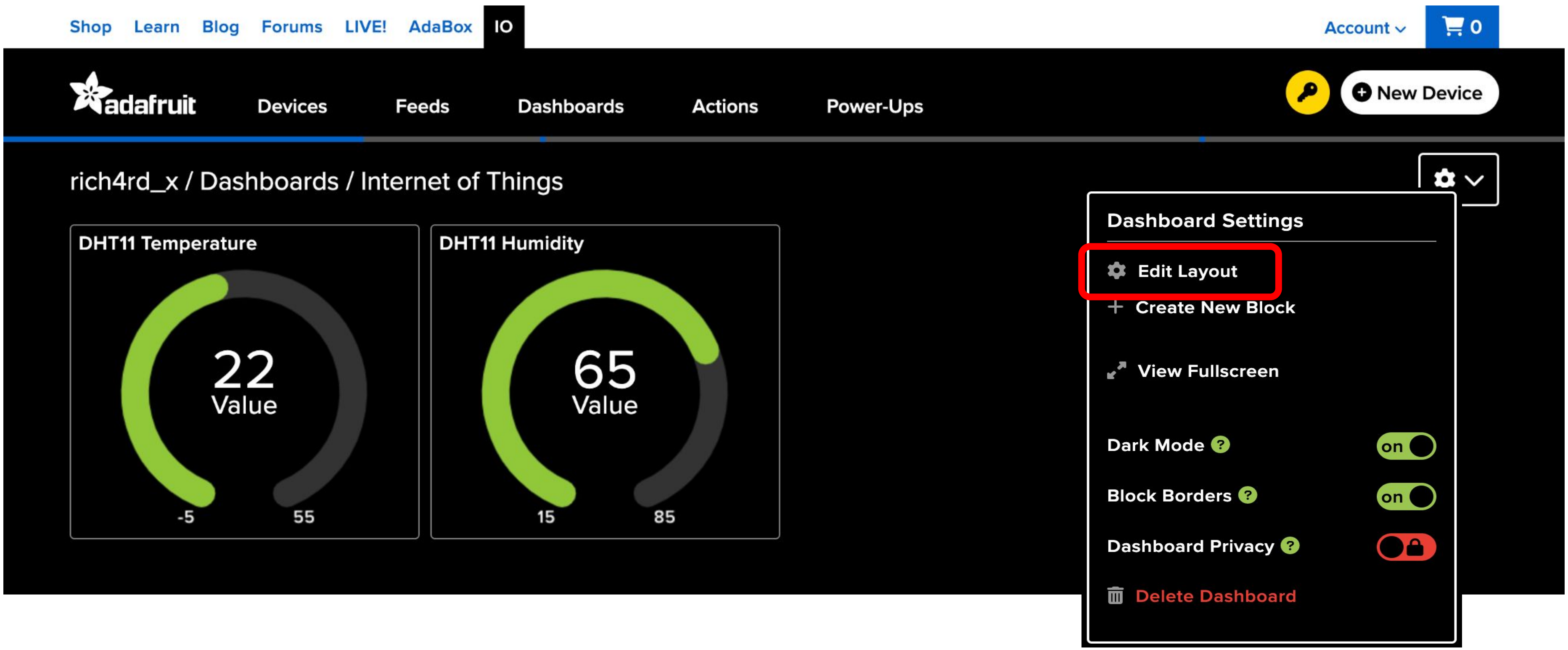
Setup Humidity Block

- Under the Block settings, setup the block as shown for the temperature gauge:
  - Block Title (optional)
  - Gauge Min Value
  - Gauge Max Value
- Once done, click “Create Block”.



Completed Adafruit.io Blocks

- Once you have created the two blocks, it should look something like this.
- Your new blocks may not be in the same layout as shown.
- However, you can change the layout to whatever that you prefer under the ‘Dashboard Settings’.



Analysing The Dashboard Data

- You have written the code to send environmental data to the cloud successfully.
- Now, the environmental data can be visualized on the dashboard easily.
- Going further, you may thinker with the Adafruit.io Dashboard settings for each gauge to set Low and High warning values.

Topic 4
Monitor Data to Trigger Control from Cloud

IoT Communication Standards

- IoT Architecture: The IoT Architecture provides a blueprint for the design and implementation of IoT systems. It defines the basic building blocks of an IoT system and the interfaces between them.
- Security Guidelines: Security is a critical aspect of IoT systems, and there are various guidelines and best practices that can help ensure the security of IoT devices and networks. These guidelines include secure boot, secure communication, and device authentication.
- Connectivity Standards: There are several connectivity standards for IoT, including Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRaWAN, Sigfox, and NB-IoT. Each of these standards has its own unique features and characteristics that make them suitable for different IoT applications.

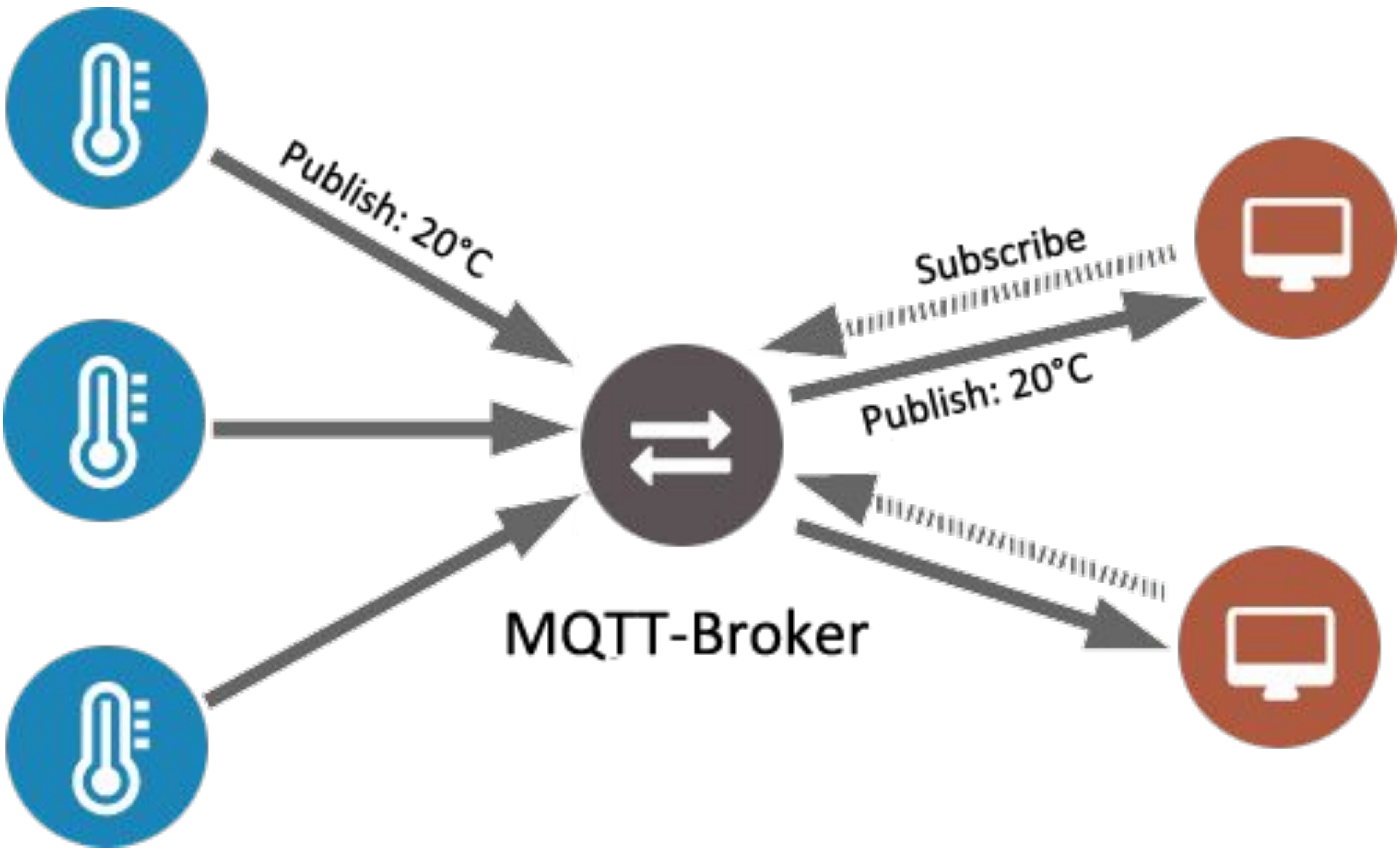
IoT Communication Standards

- Data Communication Standards: IoT devices generate large amounts of data, and there are various communication standards that can be used to transmit this data. These standards include MQTT, CoAP, REST, and WebSocket.
- Interoperability Guidelines: Interoperability is important for IoT devices to work together seamlessly. There are several guidelines and standards that can help ensure interoperability, such as the Open Connectivity Foundation (OCF), the Zigbee Alliance, and the Thread Group.

IoT Communication Standards

- Energy Efficiency Guidelines: IoT devices often operate on battery power or other low-power sources, which requires energy-efficient design and management. Guidelines for energy efficiency in IoT include minimizing power consumption, optimizing transmission schedules, and using low-power communication technologies.
- Cloud Communication Standards: Cloud computing is an important aspect of many IoT systems, and there are various communication standards that can be used to transmit data between IoT devices and the cloud. These standards include HTTP, AMQP, and MQTT.
-

MQTT Standard



MQTT Standard

- MQTT is a publish/subscribe architecture that is developed primarily to connect bandwidth and power-constrained devices over wireless networks.
- It is a simple and lightweight protocol that runs over TCP/IP sockets or WebSockets. MQTT over WebSockets can be secured with SSL.
- The publish/subscribe architecture enables messages to be pushed to the client devices without the device needing to continuously poll the server.
- The MQTT broker is the central point of communication, and it is in charge of dispatching all messages between the senders and the rightful receivers.
- A client is any device that connects to the broker and can publish or subscribe to topics to access the information. A topic contains the routing information for the broker.
- Each client that wants to send messages publishes them to a certain topic, and each client that wants to receive messages subscribes to a certain topic.
- The broker delivers all messages with the matching topic to the appropriate clients.

Key Features of MQTT

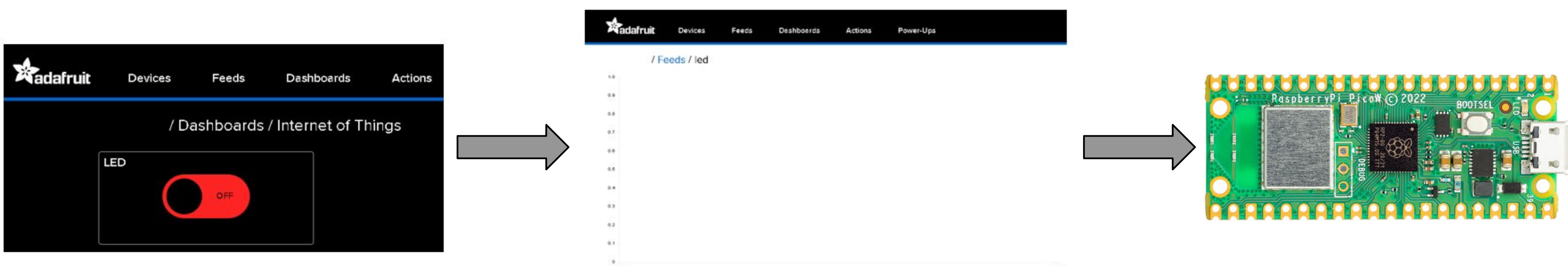
- **Pub/Sub Model:** MQTT uses a publish/subscribe (pub/sub) messaging model, where devices can publish messages to a topic, and other devices can subscribe to that topic to receive those messages.
- **QoS Levels:** MQTT supports three different quality of service (QoS) levels: 0, 1, and 2. QoS 0 delivers messages with no guarantee of delivery, while QoS 1 and QoS 2 provide increasing levels of reliability.
- **Lightweight:** MQTT is designed to be lightweight and efficient, which makes it suitable for use in resource-constrained devices, such as sensors and microcontrollers. The protocol header is only 2 bytes long, and messages can be as small as a single byte

Key Features of MQTT

- **TCP/IP Based:** MQTT uses TCP/IP as its underlying transport protocol, which provides reliable, connection-oriented communication between devices.
- **Last Will and Testament:** MQTT provides a Last Will and Testament (LWT) feature, which allows devices to specify a message that will be sent to a topic if the device unexpectedly disconnects from the network.
- **Retained Messages:** MQTT supports retained messages, which are messages that are stored on the MQTT broker and sent to any new subscribers that connect to the topic.
- **Security:** MQTT provides security features, such as authentication and encryption, to protect the privacy and integrity of IoT data.

Read Data from the Cloud

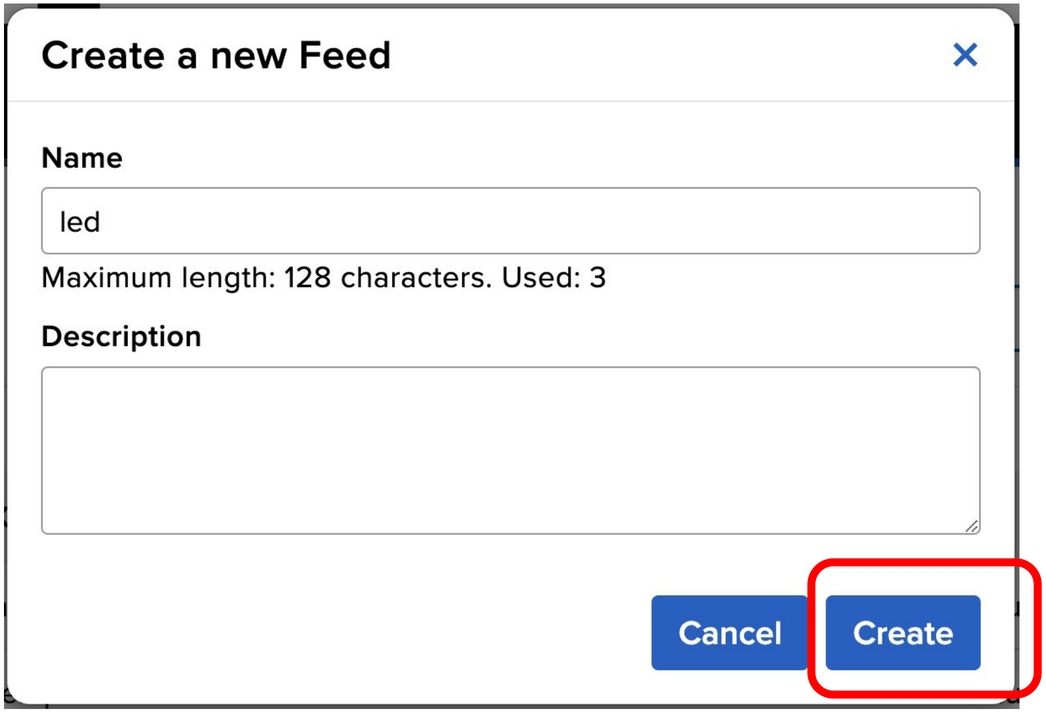
- To read data from the cloud, we could use either HTTP API or MQTT API.
- To achieve this with Adafruit.io, we will create a new Micropython code.
- This will allow us to control the Raspberry Pi Pico W on-board LED using the cloud toggle switch.



Dashboards **Feeds** **Devices**

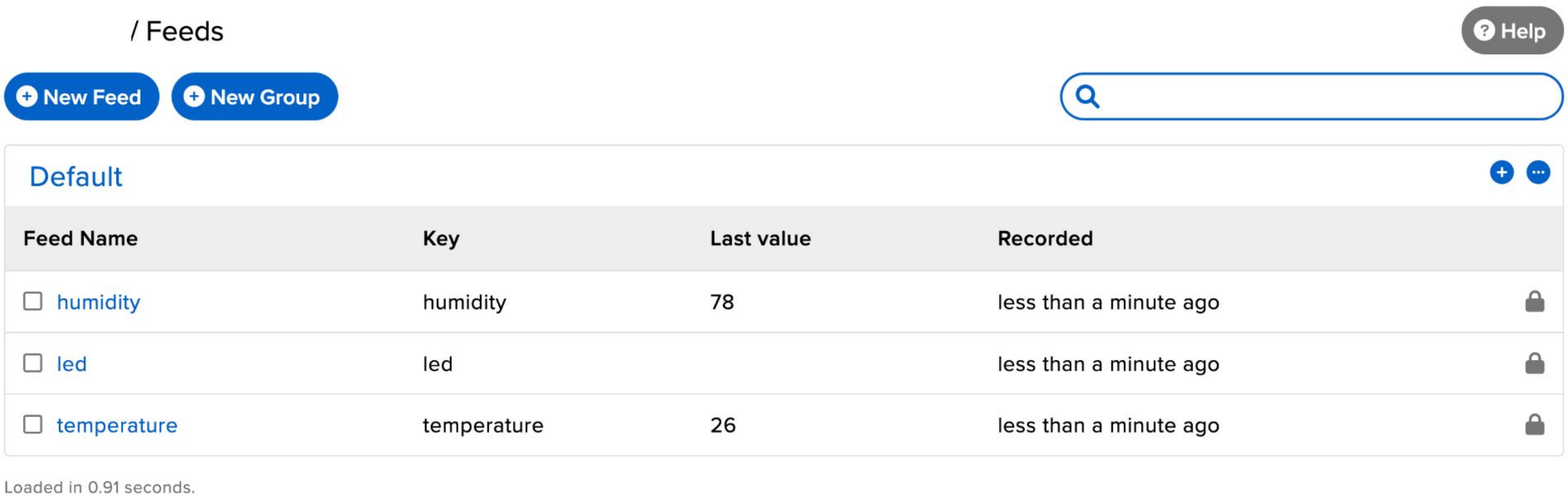
Create Adafruit.io Feed

- Follow the previous steps of creating a feed.
- After clicking on "New Feeds", you will be prompted to give the new feed a name and description (optional).
- Once done, click the "Create" button.



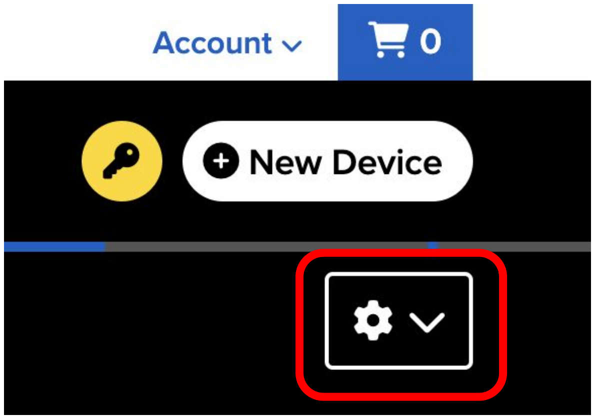
Completed Adafruit.io Feed

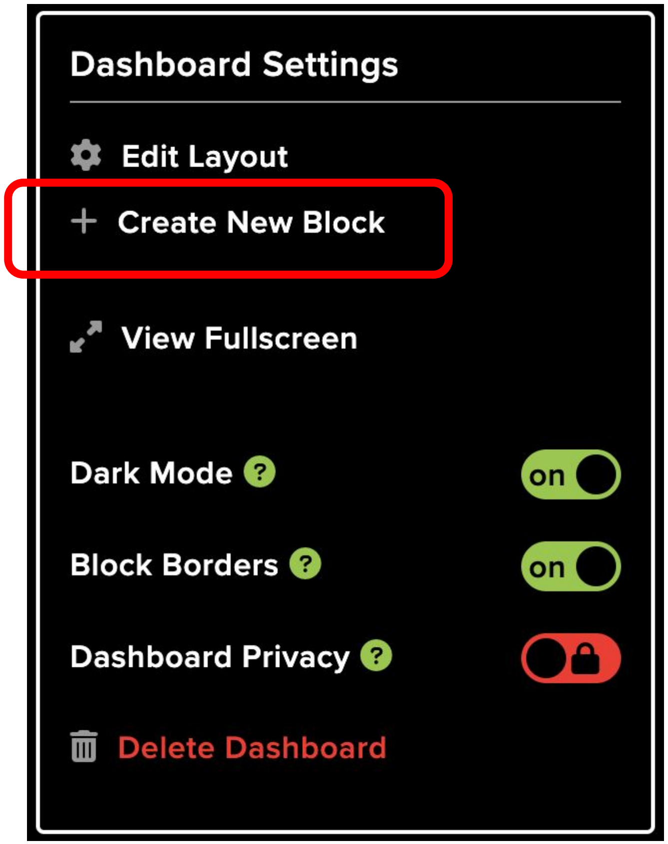
- Once you have created the led feed, it should look like this.



Setup Dashboard Blocks

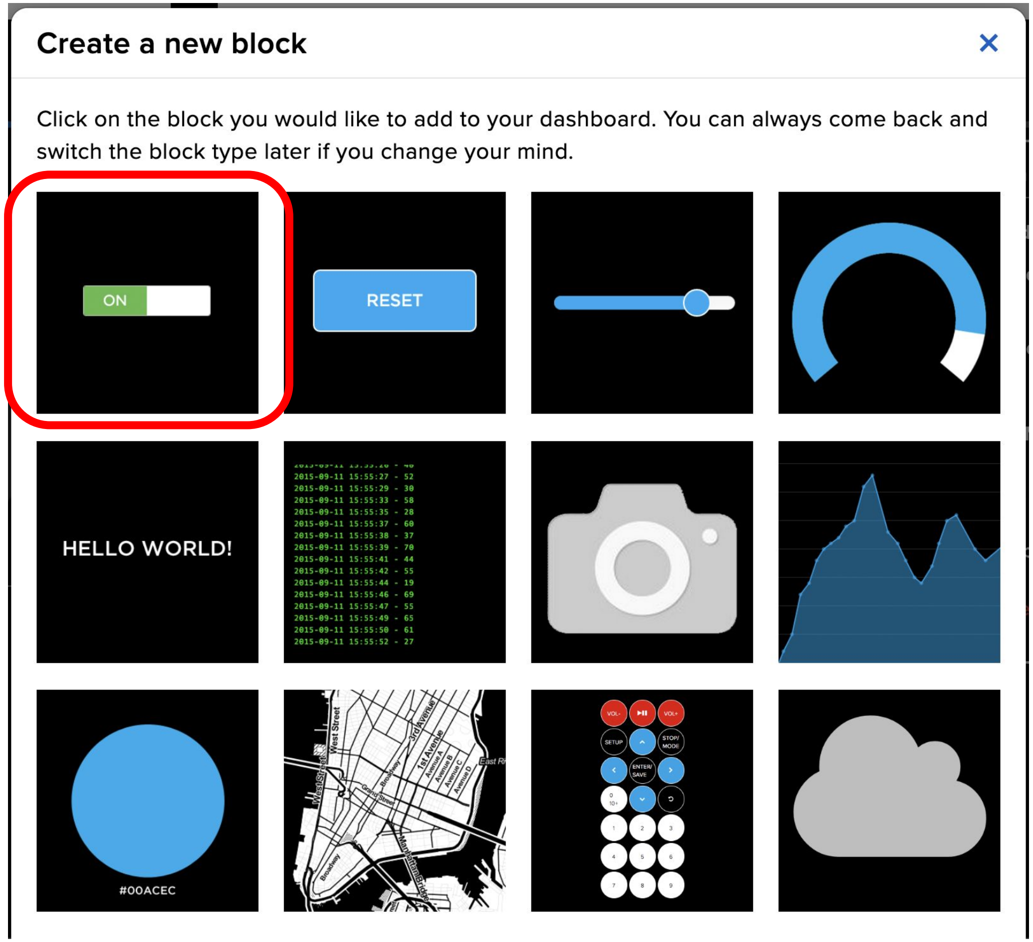
- On the existing dashboard, you will the previously created blocks.
- We will create one more block that will be used to control the in-built LED on the Raspberry Pi Pico W.
- Click on the settings dropdown icon and click “Create New Block”.





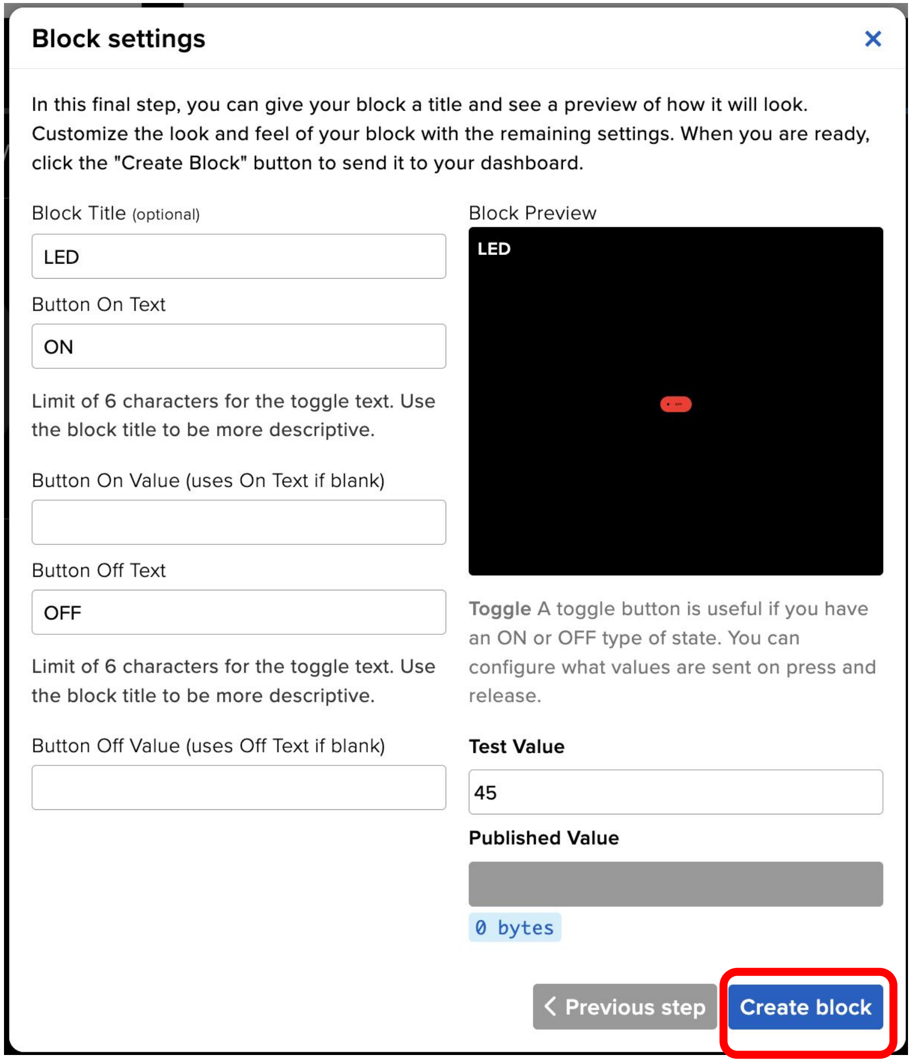
Setup Dashboard Blocks

- From the various options to choose from, choose the toggle block.
- Select the ‘led’ feed and click “Next step”.



Setup LED Block

- Under the Block settings, setup the block as shown for the led toggle block:
  - Block Title (optional)
- Once done, click “Create Block”.



Control Device via IoT

- To control the Raspberry Pi Pico W from the cloud - Control LED

```
from machine import Pin
from time import sleep
from machine import Timer
from umqtt.simple import MQTTClient
import dht
import network
import time
import sys

led = Pin('LED',Pin.OUT)

WIFI_SSID     = 'XXXXXX'
WIFI_PASSWORD = 'XXXXXX'

mqtt_client_id      = bytes('client_'+'12321', 'utf-8') # Just a random client ID

ADAFRUIT_IO_URL     = 'io.adafruit.com'
ADAFRUIT_USERNAME   = 'XXXXXX'
ADAFRUIT_IO_KEY     = 'XXXXXX'

TOGGLE_FEED_ID    = 'led'

def connect_wifi():
    wifi = network.WLAN(network.STA_IF)
    wifi.active(True)
    wifi.disconnect()
    wifi.connect(WIFI_SSID,WIFI_PASSWORD)
    if not wifi.isconnected():
        print('Connecting..')
        timeout = 0
        while (not wifi.isconnected() and timeout < 5):
            print(5 - timeout)
            timeout = timeout + 1
            time.sleep(1)
    if(wifi.isconnected()):
        print('Connected!')
```

Control Device via IoT

- To control the Raspberry Pi Pico W from the cloud - Control LED

```python
        print('Not connected!')
        sys.exit()

connect_wifi() # Connecting to WiFi Router

client = MQTTClient(client_id=mqtt_client_id,
                    server=ADAFRUIT_IO_URL,
                    user=ADAFRUIT_USERNAME,
                    password=ADAFRUIT_IO_KEY,
                    ssl=False)
try:
    client.connect()
except Exception as e:
    print('Could not connect to MQTT server {}{}'.format(type(e).__name__, e))
    sys.exit()

def cb(topic, msg): # callback function
    print('Received Data: Topic = {}, Msg = {}'.format(topic, msg))
    received_data = str(msg, 'utf-8') # Receiving data
    if received_data == "ON":
        led.on()
    if received_data == "OFF":
        led.off()

toggle_feed = bytes('{:s}/feeds/{:s}'.format(ADAFRUIT_USERNAME, TOGGLE_FEED_ID), 'u

client.set_callback(cb) # callback function
client.subscribe(toggle_feed) # subscribing to particular topic

while True:
    try:
        client.check_msg() # non blocking function
    except:
        client.disconnect()
        sys.exit()
```

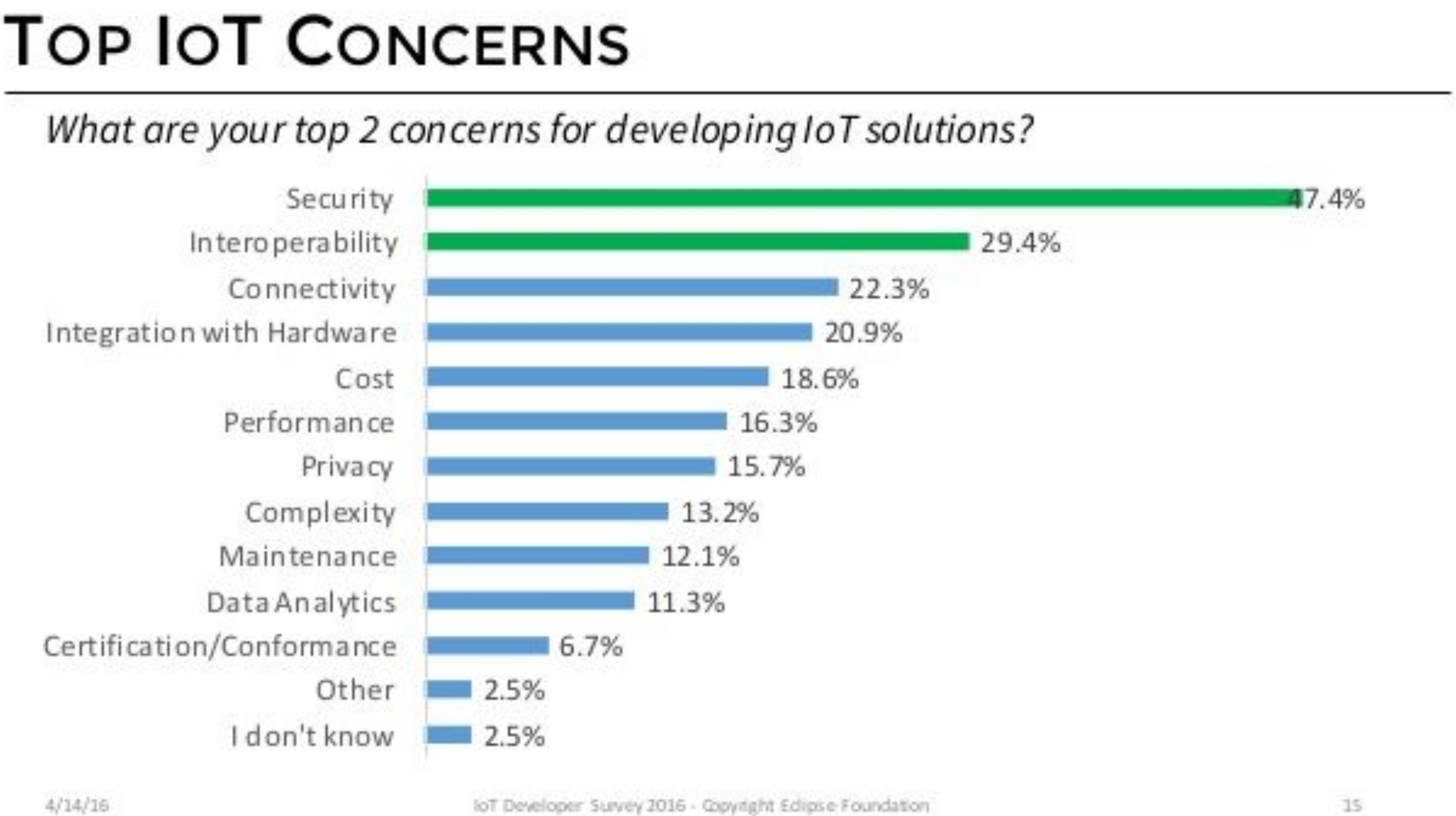
Activity: Control Device from Cloud

- Time to apply what you have learnt.
- Control the Raspberry Pi Pico W LED.
- Setup LED to 1 if the humidity is more than 80, else set the LED to 0.

What is IoT Security?

- While traditional information cybersecurity revolves around software and how it is implemented, security for IoT adds an extra layer of complexity as the cyber and the physical worlds converge.
- A wide range of operational and maintenance scenarios in the IoT space rely on end-to-end device connectivity to enable users and services to interact, login, troubleshoot, send, or receive data from devices.
- Knowing what IoT security standards to adhere to is essential, because operational technology (OT) is too important and valuable to risk in the event of breaches, disasters, and other threats.

What's the Security Concern with IoT?



What's the Security Concern with IoT?

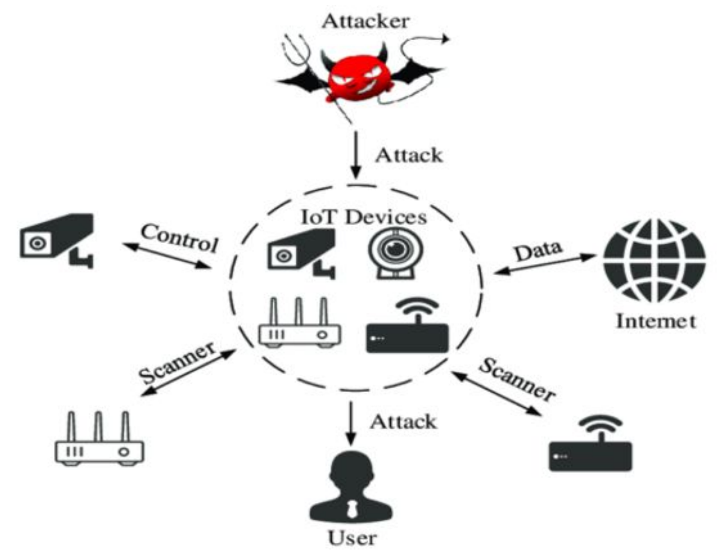
- Although IoT devices may seem too small or too specialized to be dangerous, there is real risk in what are really network-connected, general purpose computers that can be hijacked by attackers, resulting in problems beyond IoT security.
- Even the most mundane device can become dangerous when compromised over the internet—from spying with video baby monitors to interrupted services on life-saving health care equipment.
- Once attackers have control, they can steal data, disrupt delivery of services, or commit any other cybercrime they'd do with a computer.
- Attacks that compromise IoT infrastructure inflict damage, not just with data breaches and unreliable operations, but also physical harm to the facilities, or worse—to the humans operating or relying on those facilities.

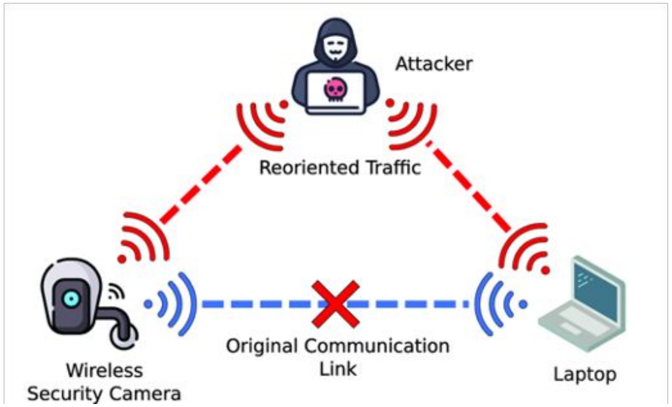
Threats of IoT Cybersecurity Attacks

- **Processes**—threats to processes both under your control, such as web services, and threats from external entities, such as users and satellite feeds, that interact with the system, but are not under the control of the application.
- **Communication**, also called data flows—threats around the communication path between devices, devices and field gateways, and device and cloud gateway.
- S**torage**—threats to temporary data queues, operating systems (OS), and image storage

Spoofing

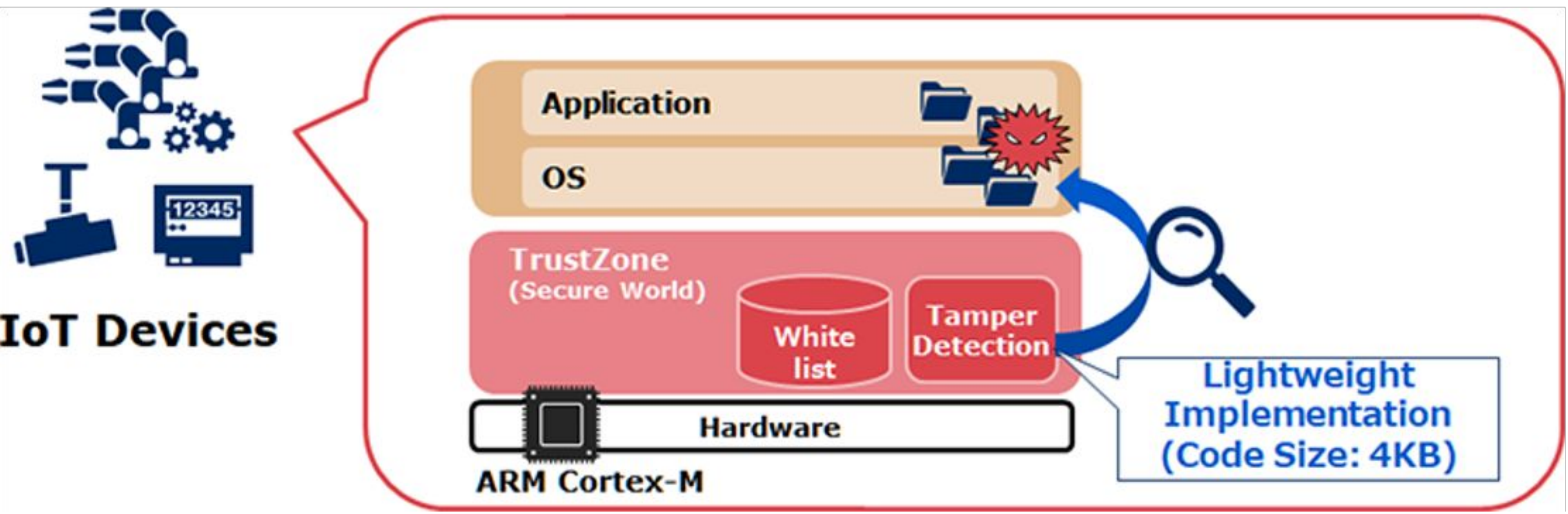
- An attacker can manipulate the state of a device anonymously.
- An attacker may intercept or partially override the broadcast and spoof the originator (often called man-in-the-middle or MitM attacks).
- An attacker can take advantages of the vulnerability of constrained or special-purpose devices. These devices, which often have one-for-all security facilities like password or PIN protection or rely on network shared key protections.
- When the shared secret to device or network (PIN, password, shared network key) is disclosed, it is possible to control the device or observe data emitted from the device.





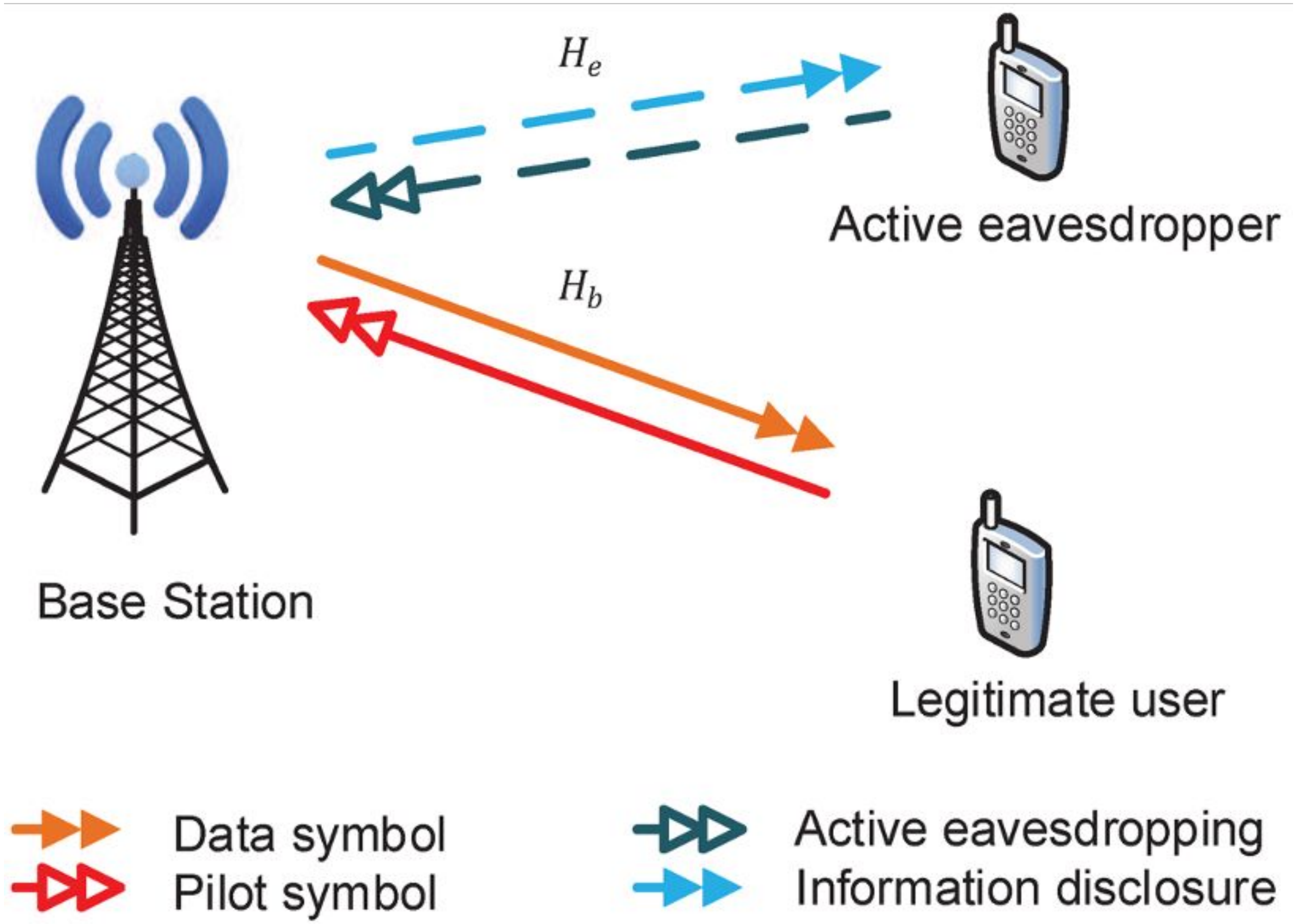
Tampering

- An attacker can tamper with any physical device—from battery drainage vulnerability or "sleep deprivation to random number generator (RNG) attacks made possible by freezing devices to reduce entropy.
- An attacker may partially or wholly replace the software running on the device, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program.



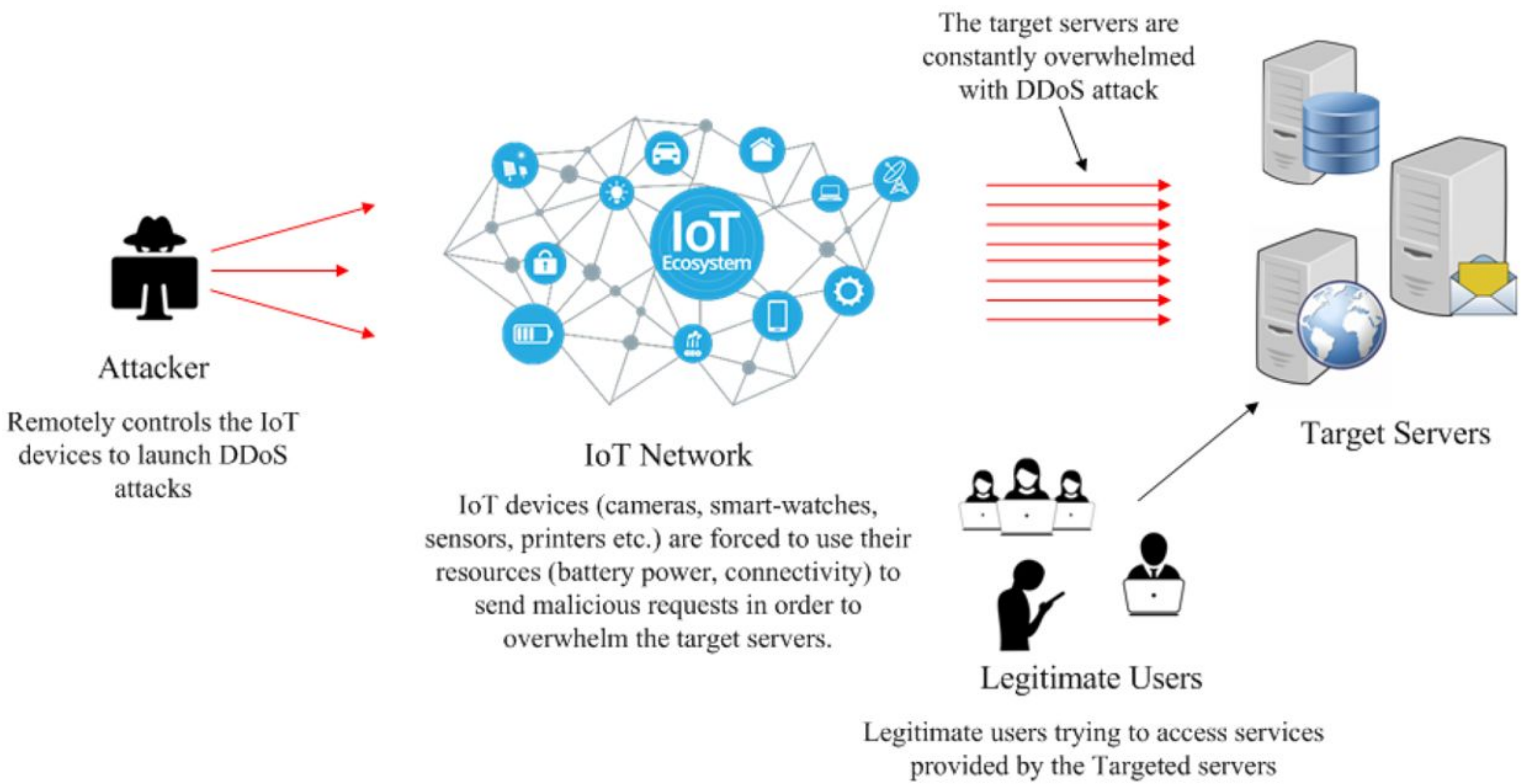
Eavesdrop

- An attacker may eavesdrop on a broadcast and obtain information without authorization or may jam the broadcast signal and deny information distribution.
- An attacker may intercept or partially override the broadcast and send false information.



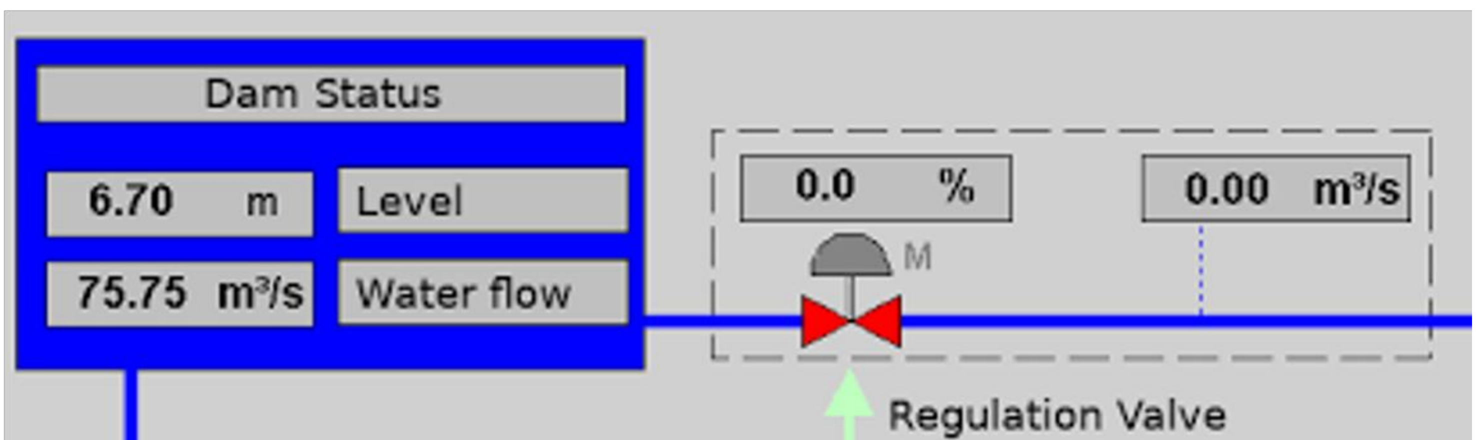
Denial of Service

- A device can be rendered incapable of functioning or communicating by interfering with radio frequencies or cutting wires.
- For example, a surveillance camera that had its power or network connection intentionally knocked out cannot report data, at all.



Elevation of Privilege

- A device that does specific function can be forced to do something else.
- For example, a valve that is programmed to open halfway can be tricked to open all the way.

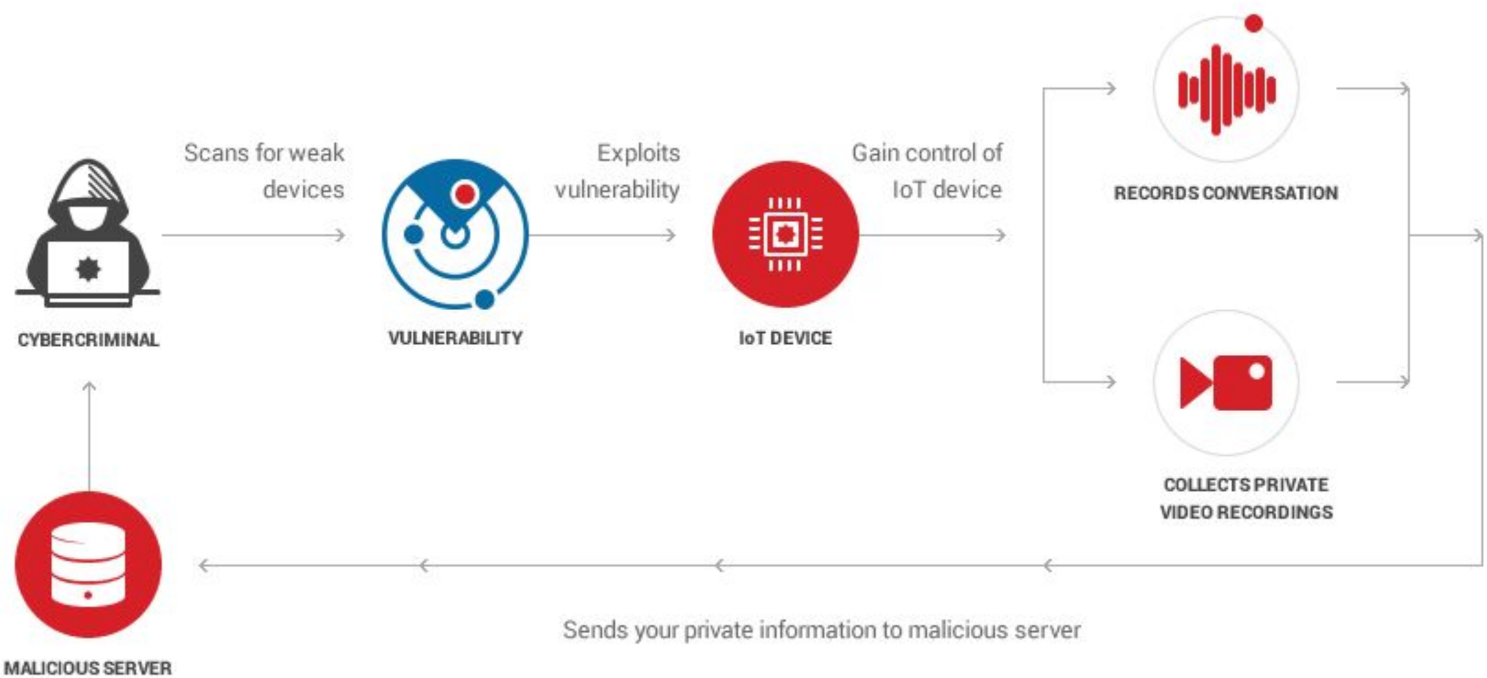


What Steps Can I Take to Secure IoT Deployments?

- Simplify security for IoT complexity
  - Integrate across teams and infrastructure to coordinate a comprehensive approach, from the physical devices and sensors to your data in the cloud.
- Prepare for IoT security specifically
  - Consider resource-constrained devices, geographic distribution of deployments, and the number of devices within an IoT security solution.
- Get smart about security analytics and remediation
  - Monitor everything connected to your IoT solution with security posture management. Stack rank the suggestions based on severity to decide what to fix first to reduce your risk. Make sure to have threat monitoring in place to get alerts and address IoT security threats quickly.
- Focus on customer and business data protection
  - By tracking all your connected data stores, admins, and other services that touch IoT, you can make sure your IoT apps are protected and your security for IoT is effective.

How to Secure IoT Devices in the Enterprise?

- Employ Device Discovery for Complete Visibility.
- Apply Network Segmentation for Stronger Defense.
- Adopt Secure Password Practices.
- Continue to Patch and Update Firmware When Available.
- Actively Monitor IoT Devices at All Times.



Summary
Q&A

Final Assessment

Written Assessment (Q&A)

This is the Written Assessment (Q&A)

Duration: 60 mins

1. The assessor will pass the questions in hardcopy to you. There are 7 questions. You need to answer all the questions.
2. This is an open book exam that must be completed individually.

.

Written Assessment (PP)

This is a Practicum Performance Assessment

Duration: 90 mins

1. The assessor will pass a practical problem in hardcopy to you.
2. You need to set up the IoT circuit, post the data to the cloud for visualization and analysis.
3. This is an open book exam that must be completed individually.

Course Feedback

https://goo.gl/R2eumq

TRAQOM Survey

- You will receive a TRAQOM link after the class.
- Please submit TRAQOM feedback and survey after the class.

Thank You!

Man Guo Chang
gc.man.sg@gmail.com